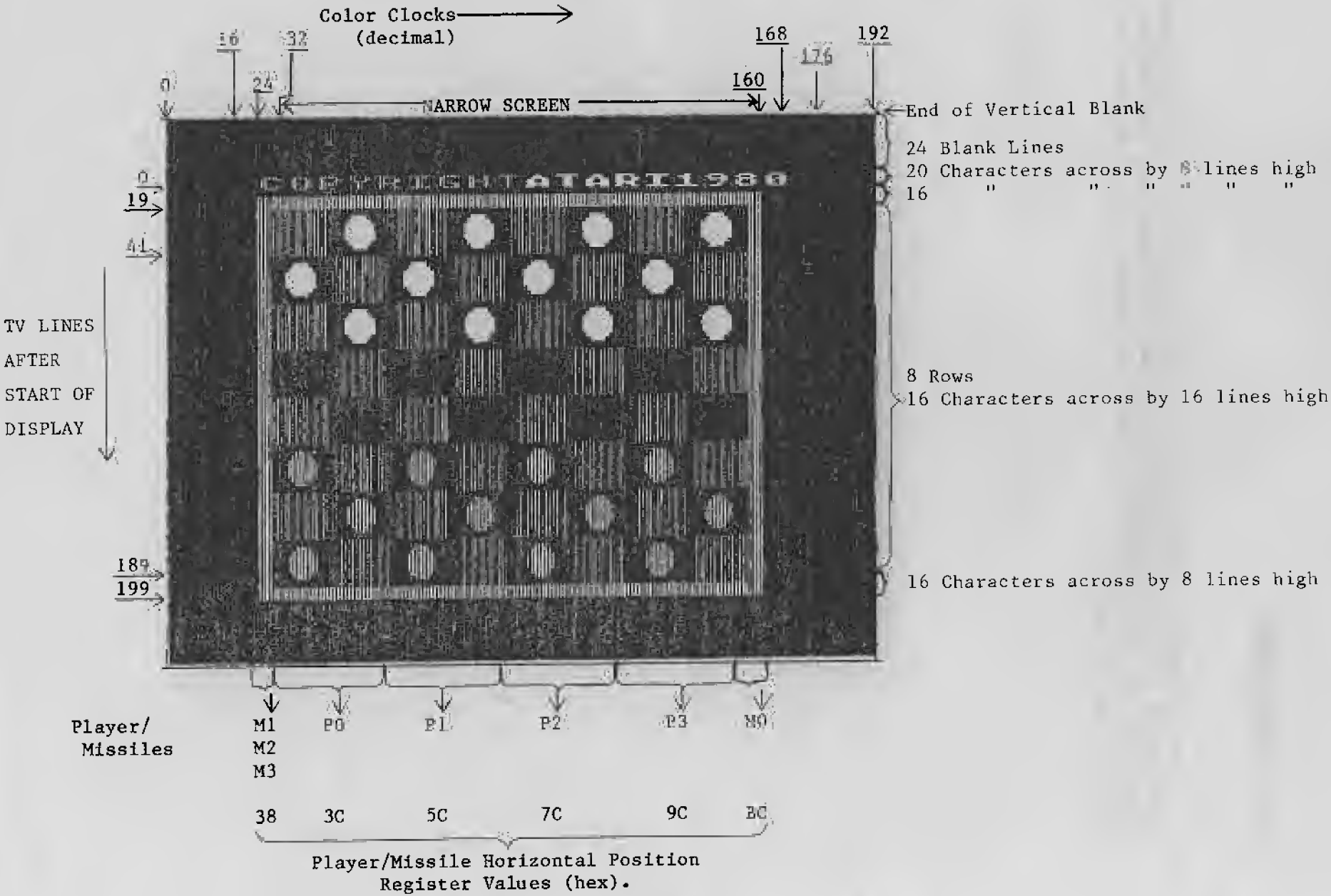
IV. SAMPLE PROGRAM

This assembly language program illustrates the use of players, missiles, and display lists. The diagram on the next page shows what the display looks like and which objects are used. The comments in the program listing describe how it works.

CHECKERS DISPLAY



```
0000                    10          TITLE "ATARI 800 CHECKERS DISPLAY BY C. SHAW 2/21/80"
                        20 ;
                        30 ;COPYRIGHT ATARI 1980
                        40 ;
                        50 ;THIS IS AN EXAMPLE OF A DISPLAY LIST WHICH USES CHARACTER MAPPING TO
                        60 ;PRODUCE THE CHECKERS AND THE TOP AND BOTTOM BORDERS OF THE BOARD.
                        70 ;PLAYERS ARE USED FOR THE RED SQUARES.  THIS GIVES 6 COLORS WITHOUT
                        80 ;CHANGING THE COLOR REGISTERS.
                        90 ;MISSILES ARE USED FOR THE LEFT AND RIGHT BORDERS
                        0100 ;THE PROGRAM STARTS AT THE LOCATION SPECIFIED BY PMB.
                        0110 ;A FEW TRICKS ARE USED TO SAVE RAM. BUT FURTHER OPTIMIZATION IS POSSIBLE
                        0120 ;THIS IS A RAM BASED PROGRAM WHICH RUNS WITH THE ASSEMBLER CARTRIDGE.  NOT A
                        0130 ;ROM CARTRIDGE.
                        0140 ;
                        0150 ;COLLEEN (ATARI 800) EQUATES
                        0160 ;
D409                    0170 CHBASE =    $D409
D400                    0180 DMACTL =    $D400
022F                    0190 SDMCTL =    $022F
D000                    0200 HPOSP0 =    $D000
D008                    0210 SIZEP0 =    $D008
02C0                    0220 PCOLR0 =    $02C0
0230                    0230 SDLSTL =    $0230
0231                    0240 SDLSTH =    $0231
D01D                    0250 GRACTL =    $D01D
D407                    0260 PMBASE =    $D407
026F                    0270 GPRIOR =    $026F
0200                    0280 VDSLST =    $0200
D40E                    0290 NMIEN  =    $D40E
                        0300 ;
                        0310 ;DISPLAY LIST EQUATES
                        0320 ;
0080                    0330 INT    =    $80       ;DISPLAY LIST INTERRUPT (BIT 7 OF NMI STATUS)
0041                    0340 JMPWT  =    $41       ;JUMP AND WAIT UNTIL END OF NEXT VERTICAL BLANK (3 BYTES)
0040                    0350 RELOAD =    $40       ;RELOAD MEM SCAN COUNTER (3 BYTES)
0020                    0360 VSC    =    $20       ;VERTICAL SCROLL ENABLE
0010                    0370 HSC    =    $10       ;HORIZONTAL SCROLL ENABLE
0001                    0380 JUMP   =    1         ;JUMP INSTRUCTION (3 BYTES)
0000                    0390 BLANK1 =    0         ;1 BLANK TV LINE
0010                    0400 BLANK2 =    $10       ;2 BLANK LINES
0020                    0410 BLANK3 =    $20       ;3
0030                    0420 BLANK4 =    $30       ;4
0040                    0430 BLANK5 =    $40       ;5
0050                    0440 BLANK6 =    $50       ;6
0060                    0450 BLANK7 =    $60       ;7
0070                    0460 BLANK8 =    $70       ;8 BLANK TV LINES
```

```
ATARI 800 CHECKERS DISPLAY BY C. SHAW 3/31/80

0000        0470          PAGE
            0480 ;
0020        0490 INTOFF =    $20       ;USED TO GET INTERNAL CODE FOR UPPER CASE ALPHANUMERICS
            0500 ;
            0510 ;INTERNAL CHARACTER CODES
            0520 ;
0000        0530 SPI     =    ' -INTOFF
0021        0540 AI      =    'A-INTOFF
0023        0550 CI      =    'C-INTOFF
0024        0560 DI      =    'D-INTOFF
0025        0570 EI      =    'E-INTOFF
0027        0580 GI      =    'G-INTOFF
0028        0590 HI      =    'H-INTOFF
0029        0600 II      =    'I-INTOFF
002F        0610 OI      =    'O-INTOFF
0030        0620 PI      =    'P-INTOFF
0032        0630 RI      =    'R-INTOFF
0034        0640 TI      =    'T-INTOFF
0039        0650 YI      =    'Y-INTOFF
0011        0660 N1I     =    '1-INTOFF
0018        0670 N8I     =    '8-INTOFF
0019        0680 N9I     =    '9-INTOFF
0010        0690 N0I     =    '0-INTOFF
            0700 ;
            0710 ;CHECKERS EQUATES
            0720 ;
            0730 ;CODES FOR SPECIAL CHECKERS CHARACTER SET
            0740 ;
0000        0750 EMPTY   =    0         ;EMPTY SQUARE
0001        0760 CHECKER=     1         ;ORDINARY CHECKER
0002        0770 KING    =    2
0003        0780 CURS    =    3         ;CURSOR (X)
0004        0790 BORDER  =    4         ;USED FOR TOP AND BOTTOM BORDERS OF BOARD
            0800 ;
0000        0810 CLP0    =    0         ;PLAYER 0 (HUMAN)
0080        0820 CLP1    =    $80       ;PLAYER 1 (COMPUTER)
00C0        0830 CLBOR   =    $C0       ;BORDER COLOR (USED TO SET UP 2 MSB'S OF CHAR)
5000        0840 PMB     =    $5000     ;PLAYER MISSILE BASE ADDRESS & PROGRAM LOCATION
```

```
ATARI 800 CHECKERS DISPLAY BY C. SHAW 3/31/80

0000        0850          PAGE
            0860 ;
            0870 ; RAM VARIABLES
            0880 ;
0000        0890          *=   PMB
5000        0900 BOARD    *=   *+32       ;CHECKER BOARD (ONLY 32 BLACK SQUARES ARE USED)
5020        0910 T0       *=   *+1        ;TEMP FOR MOVING BOARD TO MEM MAP
            0920 ;
            0930 ;PLAYER AND MISSILE GRAPHICS.
            0940 ;PLAYERS ARE USED FOR SQUARES, MISSILES FOR LEFT AND RIGHT BORDERS.
            0950 ;
5021        0960          *=   PMB+$180
5180        0970 GRM03    *=   *+$80      ;MISSILE GRAPHICS
5200        0980 GRP0     *=   *+$80      ;PLAYER 0 GRAPHICS
5280        0990 GRP1     *=   *+$80      ;PLAYER 1
5300        1000 GRP2     *=   *+$80      ;        2
5380        1010 GRP3     *=   *+$80      ;        3
            1020 ;
5400        1030 TITL     *=   *+20       ;TOP LINE OF CHARS -- ATASCII MESSAGE
5414        1040 TOPBRD   *=   *+16       ;TOP BORDER OF BOARD
5424        1050 BRDSP    *=   8*16+*     ;BOARD DISPLAY
54A4        1060 BOTBRD   *=   *+16       ;BOTTOM BORDER
```

```
ATARI 800 CHECKERS DISPLAY BY C. SHAW 3/31/80


54B4          1070        .PAGE
              1080 ;
              1090 ;GR -- SPECIAL CHECKERS CHARACTER SET (ONLY CODES 0-4 ARE USED)
              1100 ;
54B4          1110        *=   PMB+$600
              1120 GR
5600 00       1130        .BYTE 0,0,0,0,0,0,0,0 ;BLANK (0)
5601 00
5602 00
5603 00
5604 00
5605 00
5606 00
5607 00
5608 3C       1140        .BYTE $3C,$7E,$FF,$FF,$FF,$FF,$7E,$3C ;CHECKER (1)
5609 7E
560A FF
560B FF
560C FF
560D FF
560E 7E
560F 3C
5610 3C       1150        .BYTE $3C,$7E,$A5,$A5,$C3,$C3,$7E,$3C ;KING (2)
5611 7E
5612 A5
5613 A5
5614 C3
5615 C3
5616 7E
5617 3C
5618 C3       1160        .BYTE $C3,$66,$3C,$18,$18,$3C,$66,$C3 ;CURSOR (3)
5619 66
561A 3C
561B 18
561C 18
561D 3C
561E 66
561F C3
5620 00       1170        .BYTE 0,$FF,$FF,$FF,$FF,$FF,$FF,0 ;BORDER (4)
5621 FF
5622 FF
5623 FF
5624 FF
5625 FF
5626 FF
5627 00
```

ATARI 800 CHECKERS DISPLAY BY C. SHAW 3/31/80

```
5628            1180        .PAGE
                1190 ;
                1200 ;
                1210 ;DISPLAY LIST
                1220 ;
                1230 DSP
5628 70         1240        BYTE BLANK8   ;24 BLANK LINES
5629 70         1250        BYTE BLANK8
562A 70         1260        BYTE BLANK8
562B 46         1270        BYTE RELOAD+6 ;LINES 0-7.  MESSAGE LINE: 20 ACROSS X 5 COLOR X 1 LINE RESOLUTION CHARACTERS
562C 0054       1280        WORD TITL
562E 80         1290        BYTE INT+BLANK1 ;8.  INTERRUPT TO CHANGE CHARACTER BASE ADDRESS AND CHANGE TO NARROW SCREEN.
562F 06         1300        BYTE 6        ;9-16.  TOP BORDER: 16 X 5 X 1 CHARS (LAST LINE IS TOP OF 1ST ROW OF SQUARES)
5630 10         1310        BYTE BLANK2   ;17-18.  TOP OF FIRST ROW OF SQUARES
                1320 ;                     CHECKERBOARD (8 LINES OF CHARS WITH SPACES INBETWEEN - 22 LINES/SQUARE)
5631 07         1330        BYTE 7        ;19-34.  16X5X2 LINE RESOLUTION CHARS
5632 50         1340        BYTE BLANK6   ;35-40.  FIRST 3 LINES=BOTTOM OF PREVIOUS SQUARE.
5633 07         1350        BYTE 7        ;41-56
5634 50         1360        BYTE BLANK6   ;57-62.  LAST 3 LINES=TOP OF NEXT SQUARE.
5635 07         1370        BYTE 7        ;63-78
5636 50         1380        BYTE BLANK6   ;79-84
5637 07         1390        BYTE 7        ;85-100
5638 50         1400        BYTE BLANK6   ;101-106
5639 07         1410        BYTE 7        ;107-122
563A 50         1420        BYTE BLANK6   ;123-128
563B 07         1430        BYTE 7        ;129-144
563C 50         1440        BYTE BLANK6   ;145-150
563D 07         1450        BYTE 7        ;151-166
563E 50         1460        BYTE BLANK6   ;167-172
563F 07         1470        BYTE 7        ;173-188
                1480 ;                     NEXT THREE LINES ARE BOTTOM OF PREVIOUS SQUARE.
5640 10         1490        BYTE BLANK2   ;189-190.  END OF NORMAL DISPLAY (SHOULD BE ON SCREEN ON ALL TV'S).
5641 06         1500        BYTE 6        ;191-198.  BOTTOM BORDER (MAY OVERSCAN, BUT NOT ESSENTIAL TO GAME PLAY)
5642 41         1510        BYTE JMPWT    ;WAIT FOR NEXT VBLANK, THEN START OVER
5643 2856       1520        WORD DSP
                1530 ;
                1540 ;
                1550 ;DSP -- DISPLAY LIST INTERRUPT HANDLER.
                1560 ;CHANGES CHARACTER BASE AND WIDTH OF DISPLAY FOR SPECIAL CHECKERS GRAPHICS.
                1570 ;THE OS WILL CHANGE CHBASE BACK TO NORMAL DURING VERTICAL BLANK.
                1580 ;
                1590 NCHR
5645 48         1600        PHA
5646 A956       1610        LDA  #GR/256
5648 8D09D4     1620        STA  CHBASE
                1630 ;
                1640 ;INSTRUCTION FETCH DMA ENABLE, P/M 2 LINE RES, P/M DMA ENABLE, NARROW SCREEN (128 CLOCKS)
564B A92D       1650        LDA  #$2D
564D 8D00D4     1660        STA  DMACTL
5650 68         1670        PLA
5651 40         1680        RTI
```

```
ATARI 800 CHECKERS DISPLAY BY C. SHAW 3/31/80


5652            1690        PAGE
                1700 ;
                1710 ;INITIALIZATION CODE -- START EXECUTION HERE
                1720 ;
5652            1730        *=   PMB+$700
                1740 ;
                1750 ;INIT OS'S DMACTL VARIABLE
                1760 ;INSTRUCTION FETCH DMA ENABLE, P/M 2 LINE RES, P/M DMA ENABLE, STANDARD SCREEN (160 CLOCKS)
                1770 ;
5700 A92E       1780        LDA  #$2E
5702 8D2F02     1790        STA  SDMCTL
                1800 ;
                1810 ;CLEAR RAM
                1820 ;
5705 A900       1830        LDA  #0
5707 AA         1840        TAX
                1850 INITLP
5708 9D0050     1860        STA  PMB,X
570B 9D0051     1870        STA  PMB+$100,X
570E 9D0052     1880        STA  PMB+$200,X
5711 9D0053     1890        STA  PMB+$300,X
5714 9D0054     1900        STA  PMB+$400,X
5717 E8         1910        INX
5718 D0EE       1920        BNE  INITLP
                1930 ;
                1940 ;INITIALIZE MISSILE GRAPHICS FOR BORDERS
                1950 ;
571A A90E       1960        LDA  #$0E
571C A05E       1970        LDY  #$5E
571E 999451     1980 LOPZ   STA  GRM03+$14,Y
5721 88         1990        DEY
5722 D0FA       2000        BNE  LOPZ
                2010 ;
                2020 ;INITIALIZE TOP AND BOTTOM BORDERS
                2030 ;
5724 A010       2040        LDY  #16
5726 A9C4       2050        LDA  #CLBOR+BORDER
5728 991354     2060 TBLP   STA  TOPBRD-1,Y
572B 99A354     2070        STA  BOTBRD-1,Y
572E 88         2080        DEY
572F D0F7       2090        BNE  TBLP       ;CONTINUE UNTIL Y=0
                2100 ;
                2110 ;INITIALIZE PLAYER GRAPHICS FOR SQUARES (CHECKER BOARD) Y=0
                2120 ;
5731 A9F0       2130        LDA  #$F0
5733 A20A       2140 IN2    LDX  #10
5735 991852     2150 IN3    STA  GRP0+$18,Y
5738 999852     2160        STA  GRP1+$18,Y
573B 991853     2170        STA  GRP2+$18,Y
573E 999853     2180        STA  GRP3+$18,Y
                2190 ;
```

```
ATARI 800 CHECKERS DISPLAY BY C. SHAW 3/31/80


5741 48      2200         PHA
5742 A90A    2210         LDA  #$0A
5744 999851  2220         STA  GRM03+$18,Y ;REST OF MISSILE GRAPHICS
5747 68      2230         PLA
5748 C8      2240         INY
5749 CA      2250         DEX
574A 10E9    2260         BPL  IN3
574C 49FF    2270         EOR  #$FF        ;FILL IN OPPOSITE SQUARES
574E C058    2280         CPY  #88
5750 90E1    2290         BCC  IN2
5752 A008    2300         LDY  #8
             2310 ;
             2320 ;INITIALIZE PLAYER AND MISSILE POSITIONS AND COLORS
             2330 ;
5754 B9D857  2340 IN4     LDA  ITBL,Y
5757 9900D0  2350         STA  HPOSP0,Y
575A 8A      2360         TXA                ;$FF
575B 9908D0  2370         STA  SIZEP0,Y      ;$03 INDICATES 4 TIMES NORMAL SIZE (REST IS DON'T CARE)
575E B9E057  2380         LDA  ITBL1,Y
5761 99C002  2390         STA  PCOLR0,Y
5764 88      2400         DEY
5765 10ED    2410         BPL  IN4
             2420 ;
             2430 ;OS, ANTIC, POKEY INITIALIZATION
             2440 ;
5767 A928    2450         LDA  #DSP&$FF   ;DISPLAY LIST START ADDRESS (LSB)
5769 8D3002  2460         STA  SDLSTL
576C A956    2470         LDA  #DSP/256   ;MSB OF ADDRESS
576E 8D3102  2480         STA  SDLSTH
5771 A903    2490         LDA  #3         ;ENABLE PLAYER/MISSILE DMA TO GRAPHICS REGS
5773 8D1DD0  2500         STA  GRACTL
5776 A950    2510         LDA  #PMB/256   ;MSB OF ADDRESS OF PLAYER/MISSILE GRAPHICS
5778 8D07D4  2520         STA  PMBASE
577B A914    2530         LDA  #$14       ;5TH PLAYER ENABLE (USE PF3 FOR MISSILE COLOR), PF TAKES PRIO OVER PLAYERS
577D 8D6F02  2540         STA  GPRIOR     ;OS PRIORITY REG
5780 A945    2550         LDA  #NCHR&$FF ;DISPLAY LIST INTERRUPT VECTOR (LSB)
5782 8D0002  2560         STA  VDSLST
5785 A956    2570         LDA  #NCHR/256
5787 8D0102  2580         STA  VDSLST+1
578A 8E0ED4  2590         STX  NMIEN      ;X=$FF   $C0 ENABLES DISPLAY LIST & VBLANK INTERRUPTS
             2600 ;
             2610 ;INITILIZE BOARD DISPLAY
             2620 ;
578D A20B    2630         LDX  #11
             2640 BRDLP
578F A901    2650         LDA  #CHECKER+CLR0 ;HUMAN PIECES ON SQUARES 0-11
5791 9D0050  2660         STA  BOARD,X
5794 A981    2670         LDA  #CHECKER+CLR1 ;COMPUTER PIECES ON SQUARES 20-31
5796 9D1450  2680         STA  BOARD+20,X
5799 CA      2690         DEX
579A 10F3    2700         BPL  BRDLP
             2710 ;
```

```
                2720 ;MOVE COPYRIGHT MESSAGE TO MESSAGE DISPLAY LINE
                2730 ;
579C A213       2740        LDX  #19
579E BDE957     2750 IN6    LDA  COPY,X
57A1 9D0054     2760        STA  TITL,X
57A4 CA         2770        DEX
57A5 10F7       2780        BPL  IN6
                2790 ;
                2800 ;LOOP TO MOVE BOARD TO GRAPHICS AREA
                2810 ;THE CHECKERS PROGRAM LOGIC COULD BE ADDED HERE OR A VBLANK INTERRUPT COULD BE USED.
                2820 ;
                2830 LOOP
57A7 20AD57     2840        JSR  UPCHR
57AA 4CA757     2850        JMP  LOOP
                2860 ;
                2870 ;
                2880 ;
                2890 ;
                2900 ;UPCHR -- SUBROUTINE TO MOVE 32 BYTES OF CHECKER BOARD TO DISPLAY RAM.
                2910 ;
                2920 UPCHR
57AD A21F       2930        LDX  #31        ;SQUARE 31 = UPPER LEFT
57AF A000       2940        LDY  #0
                2950 UPLP1
57B1 A903       2960        LDA  #4-1       ;4 SQUARES/LINE
57B3 8D2050     2970        STA  T0
                2980 UPLP2
57B6 BD0050     2990        LDA  BOARD,X
57B9 992654     3000        STA  BRDSP+2,Y  ;FOR ROWS SHIFTED TO RIGHT
57BC BDFC4F     3010        LDA  BOARD-4,X
57BF 993454     3020        STA  BRDSP+$10,Y ;FOR ROWS SHIFTED TO LEFT
57C2 C8         3030        INY
57C3 C8         3040        INY
57C4 C8         3050        INY
57C5 C8         3060        INY
57C6 CA         3070        DEX
57C7 CE2050     3080        DEC  T0
57CA 10EA       3090        BPL  UPLP2
                3100 ;
57CC 98         3110        TYA
57CD 18         3120        CLC
57CE 6910       3130        ADC  #$10
57D0 A8         3140        TAY
57D1 8A         3150        TXA
57D2 E903       3160        SBC  #4-1       ;CARRY IS CLEAR (SUBTRACT 4)
57D4 AA         3170        TAX
57D5 B0DA       3180        BCS  UPLP1
57D7 60         3190        RTS
                3200 ;
                3210 ;
                3220 ;
                3230 ;
```

```
                3240 ;DATA
                3250 ;HORIZONTAL POSITION OF PLAYERS (SQUARES) AND MISSILES (SIDE BORDERS).
                3260 ;M0=RIGHT BORDER, M1=LEFT BORDER
                3270 ;M2 & M3 ARE PLACED WITH M1.
                3280 ;                P0, P1, P2, P3, M0, M1, M2, M3
                3290 ITBL
57D8 3C         3300          .BYTE $3C,$5C,$7C,$9C,$BC,$38,$38,$38
57D9 5C
57DA 7C
57DB 9C
57DC BC
57DD 38
57DE 38
57DF 38
                3310 ;
                3320 ;COLOR TABLE
                3330 ITBL1
57E0 34         3340          .BYTE $34,$34,$34,$34 ;4 PLAYERS (RED SQUARES)
57E1 34
57E2 34
57E3 34
57E4 36         3350          .BYTE $36       ;PF0: RED CHECKERS AND MESSAGES
57E5 88         3360          .BYTE $88       ;PF1: BLUE CHARACTERS
57E6 0E         3370          .BYTE $0E       ;PF2: WHITE CHECKERS AND MESSAGES
57E7 26         3380          .BYTE $26       ;PF3: YELLOW BORDER (CHARS & MISSILES)
57E8 00         3390          .BYTE 0         ;BK: BLACK BACKGROUND
                3400 ;
                3410 ;"COPYRIGHT ATARI 1980" MESSAGE
                3420 ;
0000            3430 OF       =     $00       ;FOR PF0 COLOR (RED)
0080            3440 OF2      =     $80       ;FOR PF2 COLOR (WHITE)
0040            3450 OF3      =     $40       ;FOR PF1 COLOR (BLUE)
                3460 TGTBL
57E9 00         3470 COPY     .BYTE SPI,CI+OF,OI+OF,PI+OF,YI+OF,RI+OF,II+OF,GI+OF,HI+OF,TI+OF
57EA 23
57EB 2F
57EC 30
57ED 39
57EE 32
57EF 29
57F0 27
57F1 28
57F2 34
57F3 A1         3480          .BYTE AI+OF2,TI+OF2,AI+OF2,RI+OF2,II+OF2,N1I+OF3,N9I+OF3,N8I+OF3,N0I+OF3
57F4 B4
57F5 A1
57F6 B2
57F7 A9
57F8 51
57F9 59
57FA 58
57FB 50
```

V. HARDWARE REGISTER LISTS

A. ADDRESS ORDER

CTIA ADDRESSES				
	WRITE		READ	
Address	Name	Description	Name	Description
D0FF ↕ D020	REPEAT AS BELOW		7 MORE TIMES	
D01F	CONSOL	Write Consol SW.Port	CONSOL	Read Consol SW. Port
D01E	HITCLR	Collision Clear		
D01D	GRACTL	Graphic Control		
D01C	VDELAY	Vert. Delay		
D01B	PRIOR	Priority Select		
D01A	COLBK	Col-lum Bkgnd		
D019	COLPF3	Color-lum of 3		
D018	COLPF2	Playfield 2		
D017	COLPF1	Playfield 1		
D016	COLPF0	Playfield 0		
D015	COLPM3	Color-lum of 3		
D014	COLPM2	Player-Missile 2	PAL	READ PAL/NTSC bits
D013	COLPM1	Player-Missile 1	TRIG3	Read Joystick
D012	COLPM0	Player-Missile 0	TRIG2	Trigger
D011	GRAFM	Graphics All Missiles	TRIG1	Buttons
D010	GRAFP3	Graphics Player 3	TRIG0	
D00F	GRAFP2	Graphics Player 2	P3PL	Read Player
D00E	GRAFP1	Graphics Player 1	P2PL	to Player
D00D	GRAFP0	Graphics Player 0	P1PL	Collisions
D00C	SIZEM	Size All Missiles	P0PL	
D00B	SIZEP3	Size Player 3	M3PL	Read Missile
D00A	SIZEP2	Size Player 2	M2PL	To Player
D009	SIZEP1	Size Player 1	M1PL	Collisions
D008	SIZEP0	Size Player 0	M0PL	
D007	HPOSM3	Horz. Posit. Missile 3	P3PF	Read Player
D006	HPOSM2	Horz. Posit. Missile 2	P2PF	To Playfield
D005	HPOSM1	Horz. Posit. Missile 1	P1PF	Collisions
D004	HPOSM0	Horz. Posit. Missile 0	P0PF	
D003	HPOSP3	Horz. Posit. Player 3	M3PF	Read Missile
D002	HPOSP2	Horz. Posit. Player 2	M2PF	To Playfield
D001	HPOSP1	Horz. Posit. Player 1	M1PF	Collisions
D000	HPOSP0	Horz. Posit. Player 0	M0PF	

ANTIC ADDRESSES

Address	WRITE		READ	
	Name	Description	Name	Description
D4FF ↕ D410	} REPEAT (AS BELOW)		15 MORE TIMES	
D40F	NMIRES	Reset NMI Interrupt Status NMI Interrupt	NMIST	NMI Interrupt Status Register
D40E	NMIEN	ENABLE		
D40D			PENV	Light Pen Register Vertical
D40C			PENH	Light Pen Register Horizontal
D40B			VCOUNT	Vertical Line Counter
D40A	WSYNC	Wait for HBLANK Synchronism		
D409	CHBASE	Character Base Address Red		
D408				
D407	PMBASE	Player-Missile Base Address Register		
D406				
D405	VSCROL	Vertical Scroll Register		
D404	HSCROL	Horizontal Scroll Register		
D403	DLISTH	Display List Pointer (High Byte)		
D402	DLISTL	Display List Poimger (Low Byte)		
D401	CHACTL	Character Control Register		
D400	DMACTL	DMA Control Register		

POKEY ADDRESSES

	WRITE		READ	
	Name	Description	Name	Description
D2FF ↕ D210	REPEAT (AS BELOW)		15 MORE TIMES	
D20F	SKCTLS	Serial Port 4 Key Control	SKSTAT	Serial Port 4 Key Status Register
D20E	IRQEN	IRQ Interrupt Enable	IRQST	IRQ Interrupt Status Register
D20D	SEROUT	Serial Port Output Reg.		Serial Port Input Register
D20C				
D20B	POTGO	Start Pot Scan Sequence		Vertical Line
D20A	SKRES	Reset Status (SKSTAT)	RANDOM	Random Numb Generator
D209	STIMER	Start Timers	KBCODE	Keyboard Code
D208	AUDCTL	Audio Control	ALLPOT	Read 8 Line Pot Port State
D207	AUDC4	Audio Channel 4 Control	POT 7	
D206	AUDF4	Audio Channel 4 Frequency	POT 6	
D205	AUDC3	Audio Channel 4 Control	POT 5	Read the value of each POT
D204	AUDF3	Audio Channel 3 Frequency	POT 4	
D203	AUDC2	Audio Channel 2 Control	POT 3	
D202	AUDF2	Audio Channel 2 Frequency	POT 2	
D201	AUDC1	Audio Channel 1 Control	POT 1	
D200	AUDF1	Audio Channel 1 Frequency	POT 0	

PIA ADDRESSES

Address	WRITE		READ	
	Name	Description	Name	Description
D3FF D304	Repeat as shown below many times			
D303	PBCTL	PORT B CONTROL	PBCTL	Same as write
D302	PACTL	PORT A CONTROL	PACTL	Same as write
D301	PORTB	Direction Register If PBCTL Bit 2=0 (otherwise)	PORTB	Same as write
	PORTB	Jack 2 & Jack 3 If Direction Bits Are 1 *	PORTB	Jack 2 & Jack 3 If Direction Bits Are 0 *
D300	PORTA	Direction Register If PACTL Bit 2=0 (Otherwise)	PORTA	Same as write
	PORTA	Jack 0 & Jack 1 If Direction Bits Are 1 *	PORTA	Jack 0 & Jack 1 If Direction Bits Are 0 *

* NOTE: Output data is retained in Jack Output Registers. If direction bits are true, a read of the jacks will read old data from these registers.

B. ALPHABETICAL ORDER

Hardware Register				OS Shadow		
		Address			Address	
Name	Description	Hex	Dec	Name	Hex	Dec
ALLPOT	Read 8 line Pot Port State	D208	53768			
AUDC1	Audio Channel 1 Control	D201	53761			
AUDC2	Audio Channel 2 Control	D203	53763			
AUDC3	Audio Channel 3 Control	D205	53765			
AUDC4	Audio Channel 4 Control	D207	53767			
AUDCTL	Audio Control	D208	53768			
AUDF1	Audio Channel 1 Frequency	D200	53760			
AUDF2	Audio Channel 2 Frequency	D202	53762			
AUDF3	Audio Channel 3 Frequency	D204	53764			
AUDF4	Audio Channel 4 Frequency	D206	53766			
CHACTL	Character Control	D401	54273	CHART	2F3	755
CHBASE	Character base address	D409	54281	CHBAS	2F4	756
COLBK	Color-Luminance of Background	D01A	53274	COLOR4	2C8	712
COLPF0	Color Luminance of Playfield 0	D016	53270	COLOR0	2C4	708
COLPF1	Color Luminance of Playfield 1	D017	53271	COLOR1	2C5	709
COLPF2	Color Luminance of Playfield 2	D018	53272	COLOR2	2C6	710
COLPF3	Color Luminance of Playfield 3	D019	53273	COLOR3	2C7	711
COLPM0	Color Luminance of Player-Missile 0	D012	53266	PCOLR0	2C0	704
COLPM1	Color Luminance of Player-Missile 1	D013	53267	PCOLR1	2C1	705
COLPM2	Color Luminance of Player-Missile 2	D014	53268	PCOLR2	2C2	706
COLPM3	Color Luminance of Player-Missile 3	D015	53269	PCOLR3	2C3	707
CONSOL	Console Switch Port	D01F	53279	Set to 8 during VBLANK		
DLISTH	Display List Pointer (high byte)	D403	54275	SDLSTH	231	561
DLISTL	Display List Pointer (low byte)	D402	54274	SDLSTL	230	560
DMACTL	Direct Memory Access (DMA) Control	D400	54272	SDMCTL	22F	559
GRACTL	Graphic Control	D01D	53277			
GRAFM	Graphics for all Missiles	D011	53265			
GRAFP0	Graphics for Player 0	D00D	53261			
GRAFP1	Graphics for Player 1	D00E	53262			
GRAFP2	Graphics for Player 2	D00F	53263			
GRAFP3	Graphics for Player 3	D010	53264			
HITCLR	Colission Clear	D01E	53278			
HPOSM0	Horizontal Position of Missile 0	D004	53252			
HPOSM1	Horizontal Position of Missile 1	D005	53253			
HPOSM2	Horizontal Position of Missile 2	D006	53254			
HPOSM3	Horizontal Position of Missile 3	D007	53255			
HPOSP0	Horizontal Position of Player 0	D000	53248			
HPOSP1	Horizontal Position of Player 1	D001	53249			
HPOSP2	Horizontal Position of Player 2	D002	53250			
HPOSP3	Horizontal Position of Player 3	D003	53251			
HSCROL	Horizontal Scroll	D404	54276			
IRQEN	Interrupt Request (IRQ) Enable	D20E	53774	POKMSK	10	16
IRQST	IRQ Status	D20E	53774			
KBCODE	Keyboard Code	D209	53769	CH	2FC	764
M0PF	Missile 0 to Playfield Collisions	D000	53248			
M0PL	Missile 0 to Player Collisions	D008	53256			
M1PF	Missile 1 to Playfield Collisions	D001	53249			
M1PL	Missile 1 to Player Collisions	D009	53257			
M2PF	Missile 2 to Playfield Collisions	D002	53250			
M2PL	Missile 2 to Player Collisions	D00A	53258			

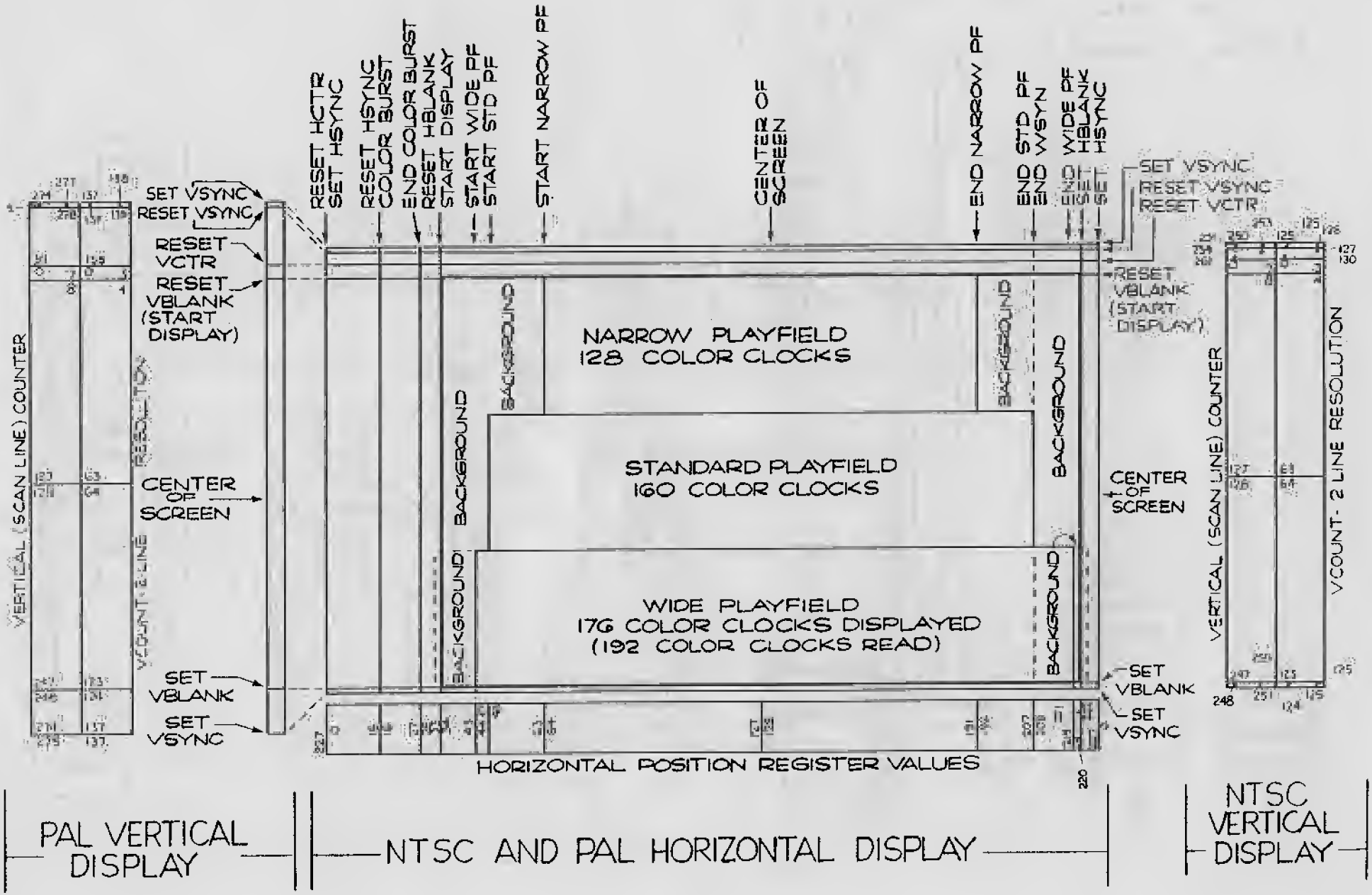
Hardware Register				OS Shadow		
		Address			Address	
Name	Description	Hex	Dec	Name	Hex	Dec
M3PF	Missile 3 to Playfield Collisions	D003	53251			
M3PL	Missile 3 to Player	D00B	53259			
NMIEN	Non-Maskable Interrupt (NMI) Enable	D40E	54286	Set to $40 by IRQ code		
NMIRES	NMI reset	D40F	54287	written to by NMI code		
NMIST	NMI Status	D40F	54287	read by NMI code		
POPF	Player 0 to Playfield Collisions	D004	53252			
POPL	Player 0 to Player Collisions	D00C	53260			
P1PF	Player 1 to Playfield Collisions	D005	53253			
P1PL	Player 1 to Player Collisions	D00D	53261			
P2PF	Player 2 to Playfield Collisions	D006	53254			
P2PL	Player 2 to Player Collisions	D00E	53262			
P3PF	Player 3 to Playfield Collisions	D007	53255			
P3PL	Player 3 to Player Collisions	D00F	53263			
PACTL	Port A Control	D302	54018	Set to $3C by IRQ Code		
PAL	PAL/NTSC indictor	D014	53268			
PBCTL	Port B Control	D303	54019	Set to $3C by IRQ Code		
PENH	Light Pen Horizontal Position	D40C	54284	LPENH	234	564
PENV	Light Pen Vertical Position	D40D	54285	LPENV	235	565
PMBASE	Player Missile Base Address	D407	54279			
PORTA	Port A	D300	54016	STICK0,1	278,279	632,633
PORTB	Port A	D301	54017	STICK2,3	27A,27B	634,635
POT0	Pot 0	D200	53760	PADDL0	270,	624
POT1	Pot 1	D201	53761	PADDL1	271	625
POT2	Pot 2	D202	53762	PADDL2	272	626
POT3	Pot 3	D203	53763	PADDL3	273	627
POT4	Pot 4	D204	53764	PADDL4	274	628
POT5	Pot 5	D205	53765	PADDL5	275	629
POT6	Pot 6	D206	53766	PADDL6	276	630
POT7	Pot 7 (right paddle controller)	D207	53767	PADDL7	277	631
POTGO	Start POT Scan Sequence	D20B	53771	WRITTEN DURING VBLANK		
PRIOR	Priority Select	D01B	53275	GPRIOR	26F	623
RANDOM	Random number generator	D20A	53770			
SERIN	Serial Port Input	D20E	53774			
SEROUT	Serial Port output	D20D	53773			
SIZEM	Sizes for all missiles	D00C	53260			
SIZEP0	Size of Player 0	D008	53256			
SIZEP1	Size of Player 1	D009	53257			
SIZEP2	Size of Player 2	D00A	53258			
SIZEP3	Size of Player 3	D00B	53259			
SKCTL	Serial Port Control	D20F	53775	SSKCTL	232	562
SKREST	Reset Serial Port Status (SKSTAT)	D20A	53770			
SKSTAT	Serial Port Status	D20F	53775			
STIMER	Start Timer	D209	53769			
TRIG0	Joystick Controller Trigger 0	D010	53264	STRIG0	284	644
TRIG1	Joystick Controller Trigger 1	D011	53265	STRIGL	285	645
TRIG2	Joystick Controller Trigger 2	D012	53266	STRIG2	286	646
TRIG3T	Joystick Controller Trigger 3	D013	53267	STRIG3	287	647
VCOUNT	Vertical Line Counter	D40B	54283			
VDELAY	Verical Delay	D01C	54276			
VSCROL	Vertical Scroll	D405	54277			
WSYNC	Wait for Horizontal Sync	D40A	54282	Used by keyboard click routine		

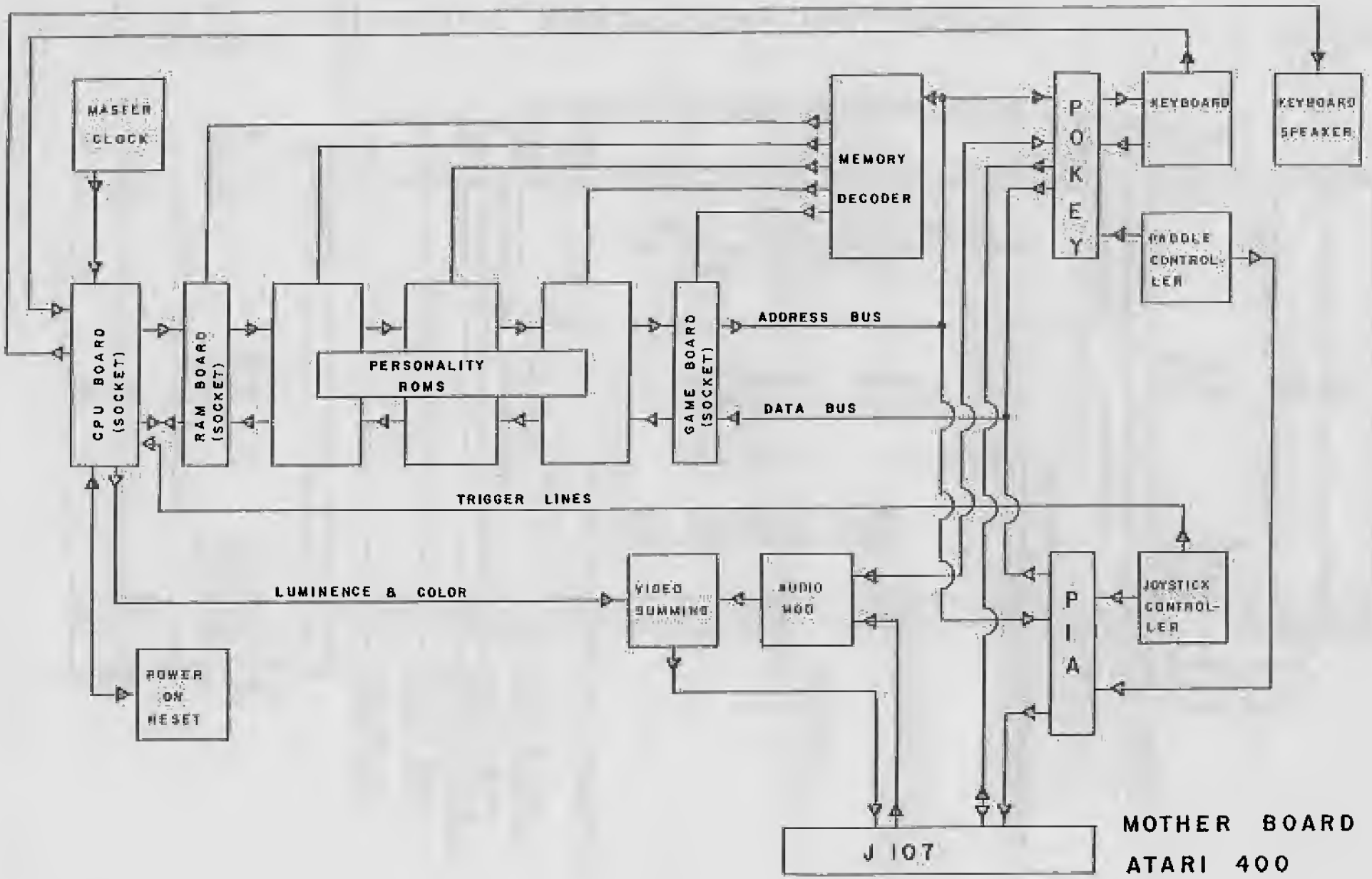
VI. FIGURES

A. MEMORY MAP

ADDRESS	FUNCTION	SIZE
FFFF D800	Operating System And Math Routines	10K
D000-D7FF CFFF	Hardware Addresses Reserved for Future O.S. expansion	2K 4K
BFFF 8000	ROM Cartridge (Colleen left and right slot and Candy single slot all address to this space)	16K
7FFF 2000	RAM Expansion *	
1FFF	RAM initially supplied in the product	8K

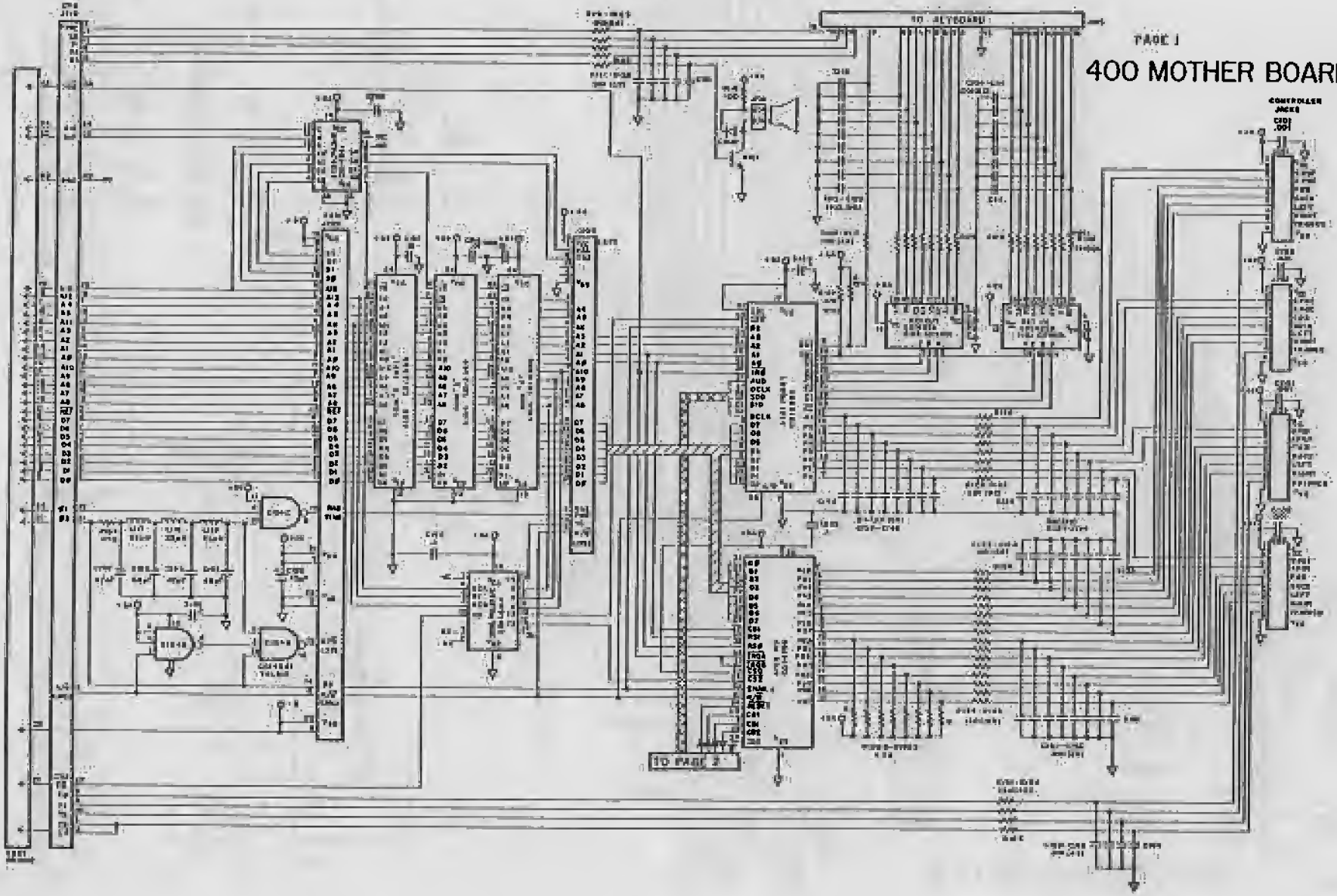
* RAM expansion can actually exten to BFFF. However, the ROM cartridges will deselect the RAM. Deselection occurs on 8K boundaries. Atari 400 units are RAM expandable only at the factory. They can accept RAM up to 2FFF (16K) when fully extended.


RESET HCTR
SET HSYNC
RESET HSYNC
COLOR BURST
END COLOR BURST
RESET HBLANK
START DISPLAY
START WIDE PF
START STD PF
START NARROW PF
CENTER OF SCREEN
END NARROW PF
END STD PF
END WSYN
WIDE PF HBLANK
SET HSYNC
SET VSYNC
RESET VSYNC
RESET VCTR
RESET VBLANK (START DISPLAY)
SET VSYNC
RESET VSYNC
RESET VCTR
RESET VBLANK (START DISPLAY)
NARROW PLAYFIELD
128 COLOR CLOCKS
STANDARD PLAYFIELD
160 COLOR CLOCKS
WIDE PLAYFIELD
176 COLOR CLOCKS DISPLAYED
(192 COLOR CLOCKS READ)
BACKGROUND
CENTER OF SCREEN
SET VBLANK
SET VSYNC
HORIZONTAL POSITION REGISTER VALUES
VERTICAL (SCAN LINE) COUNTER
VCOUNT-2 LINE RESOLUTION
VERTICAL (SCAN LINE) COUNTER
VCOUNT-2 LINE RESOLUTION
PAL VERTICAL DISPLAY
NTSC AND PAL HORIZONTAL DISPLAY
NTSC VERTICAL DISPLAY

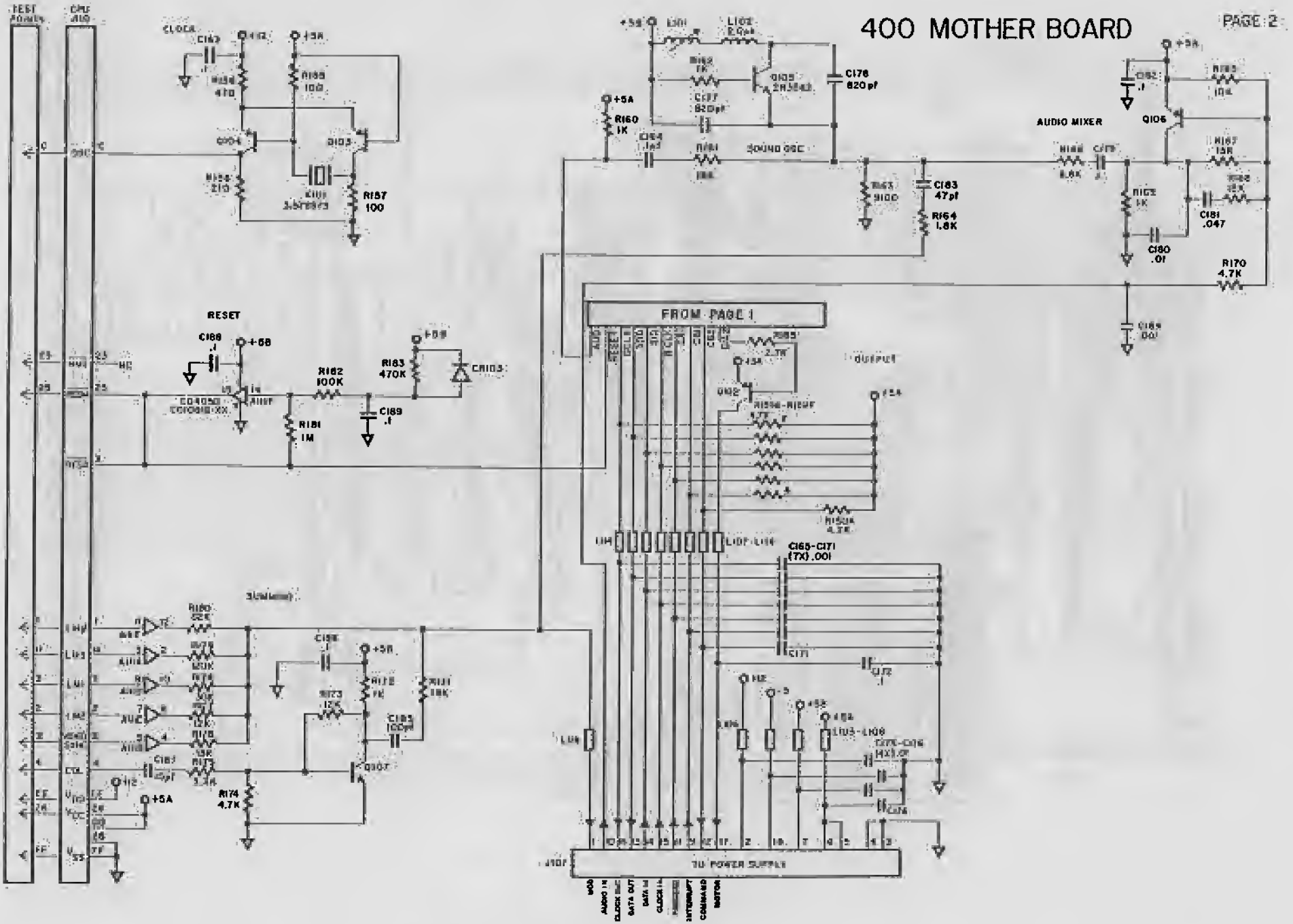

MASTER CLOCK
MEMORY DECODER
POKEY
KEYBOARD
KEYBOARD SPEAKER
PADDLE CONTROL-LER
CPU BOARD (SOCKET)
RAM BOARD (SOCKET)
PERSONALITY ROMS
GAME BOARD (SOCKET)
ADDRESS BUS
DATA BUS
TRIGGER LINES
LUMINENCE & COLOR
VIDEO SUMMING
AUDIO MOD
PIA
JOYSTICK CONTROL-LER
POWER ON RESET
J 107
MOTHER BOARD
ATARI 400

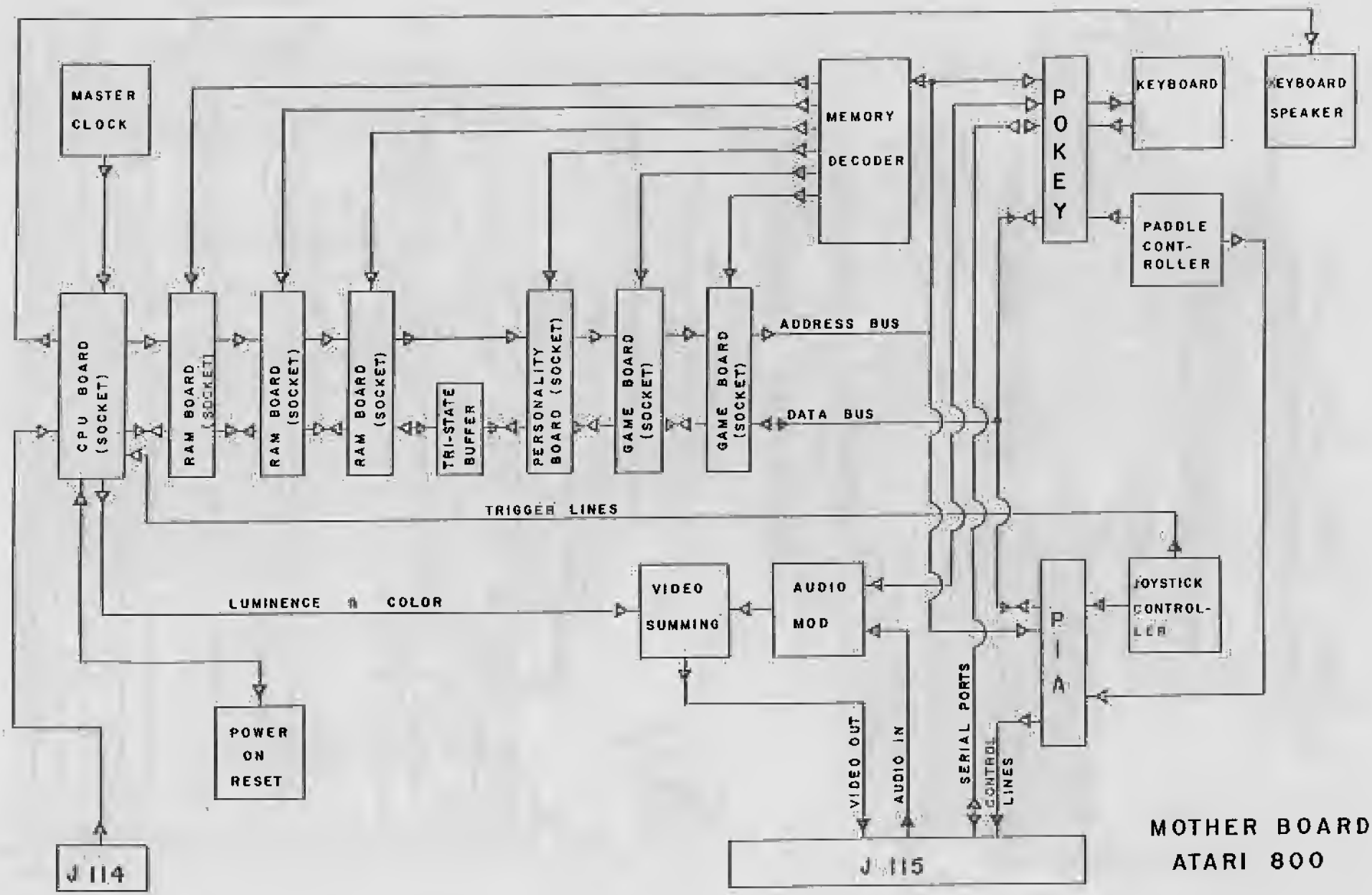
PAGE 1

400 MOTHER BOARD



400 MOTHER BOARD

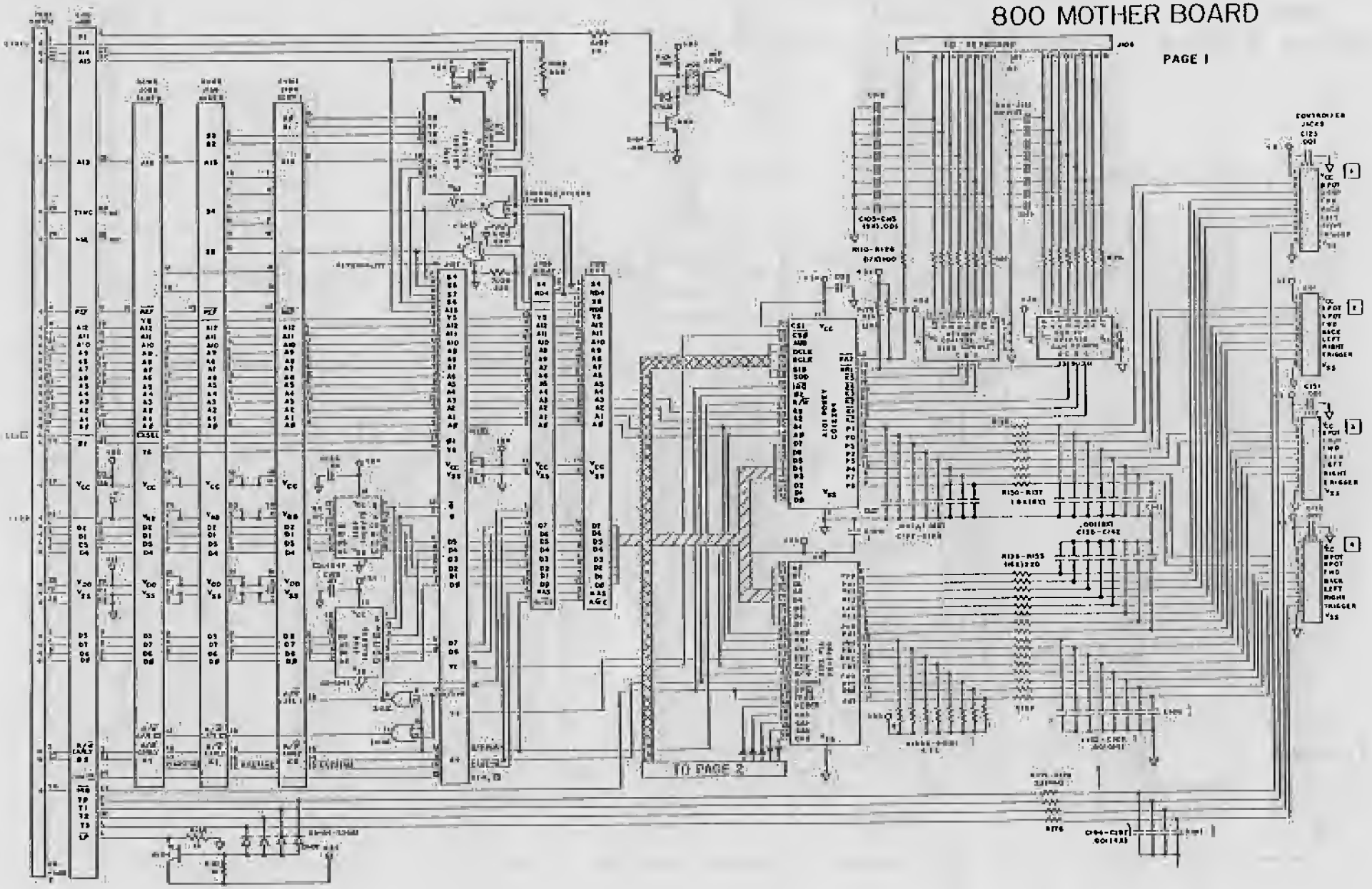




MOTHER BOARD
ATARI 800

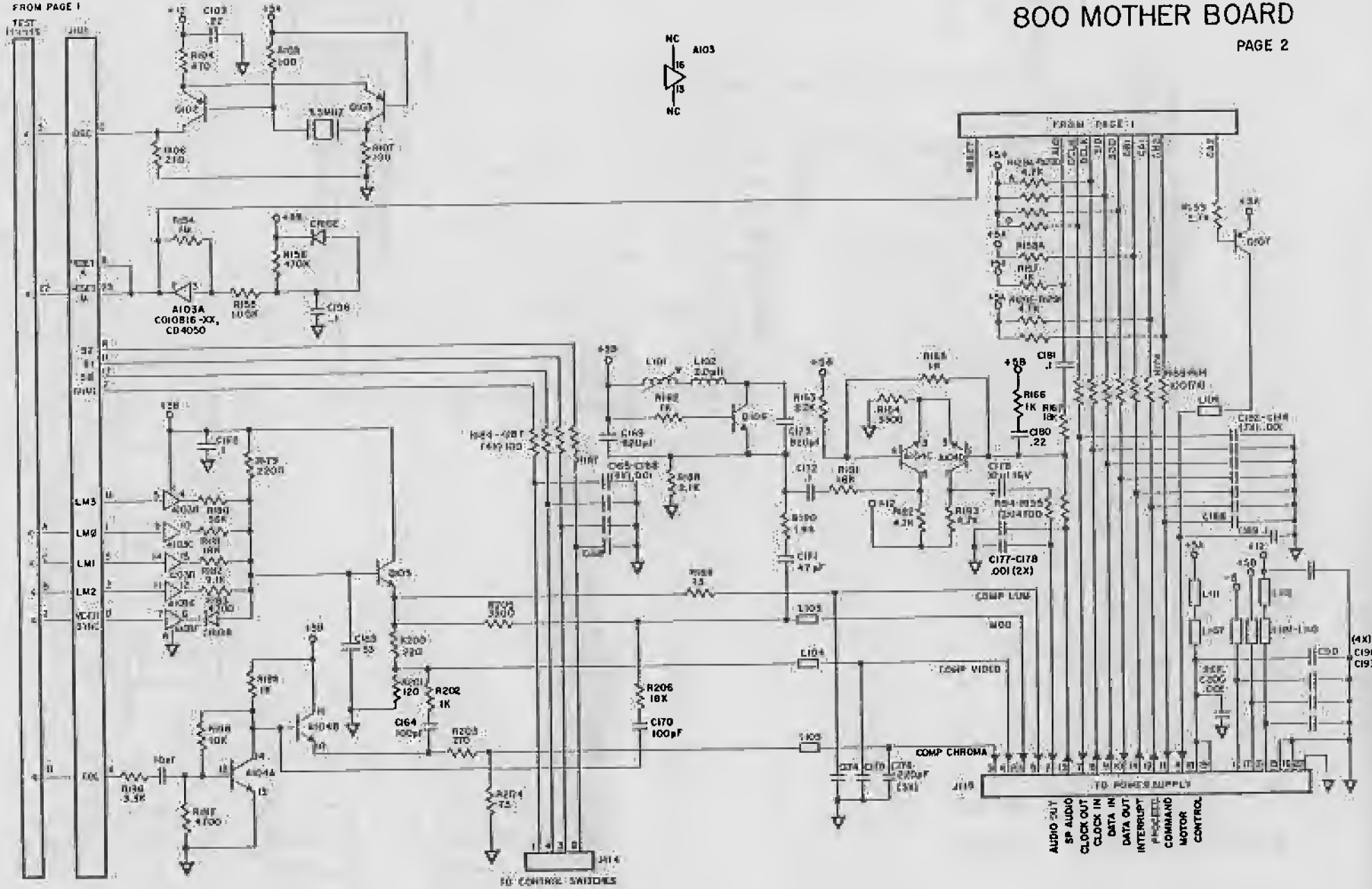
800 MOTHER BOARD

PAGE 1

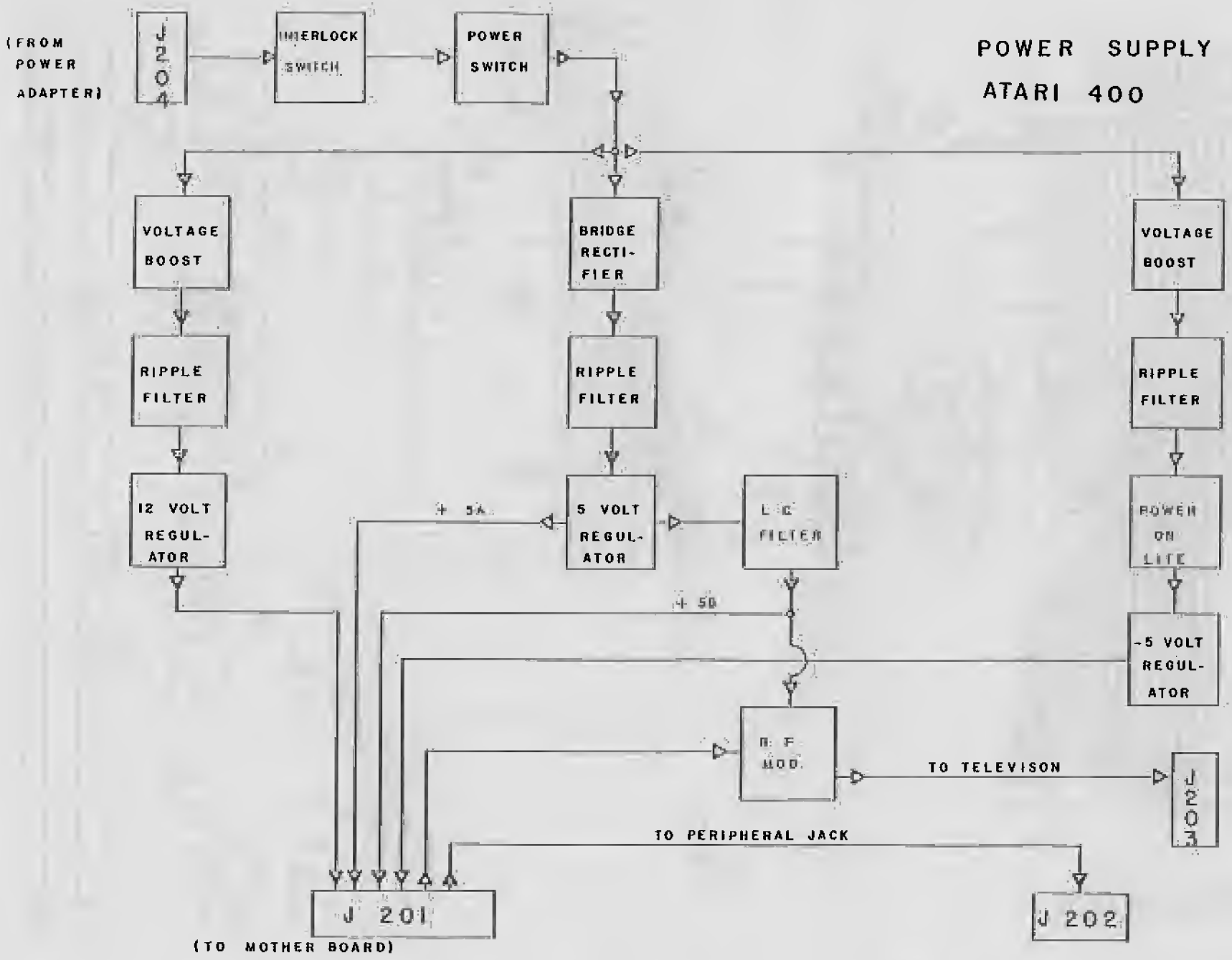


800 MOTHER BOARD

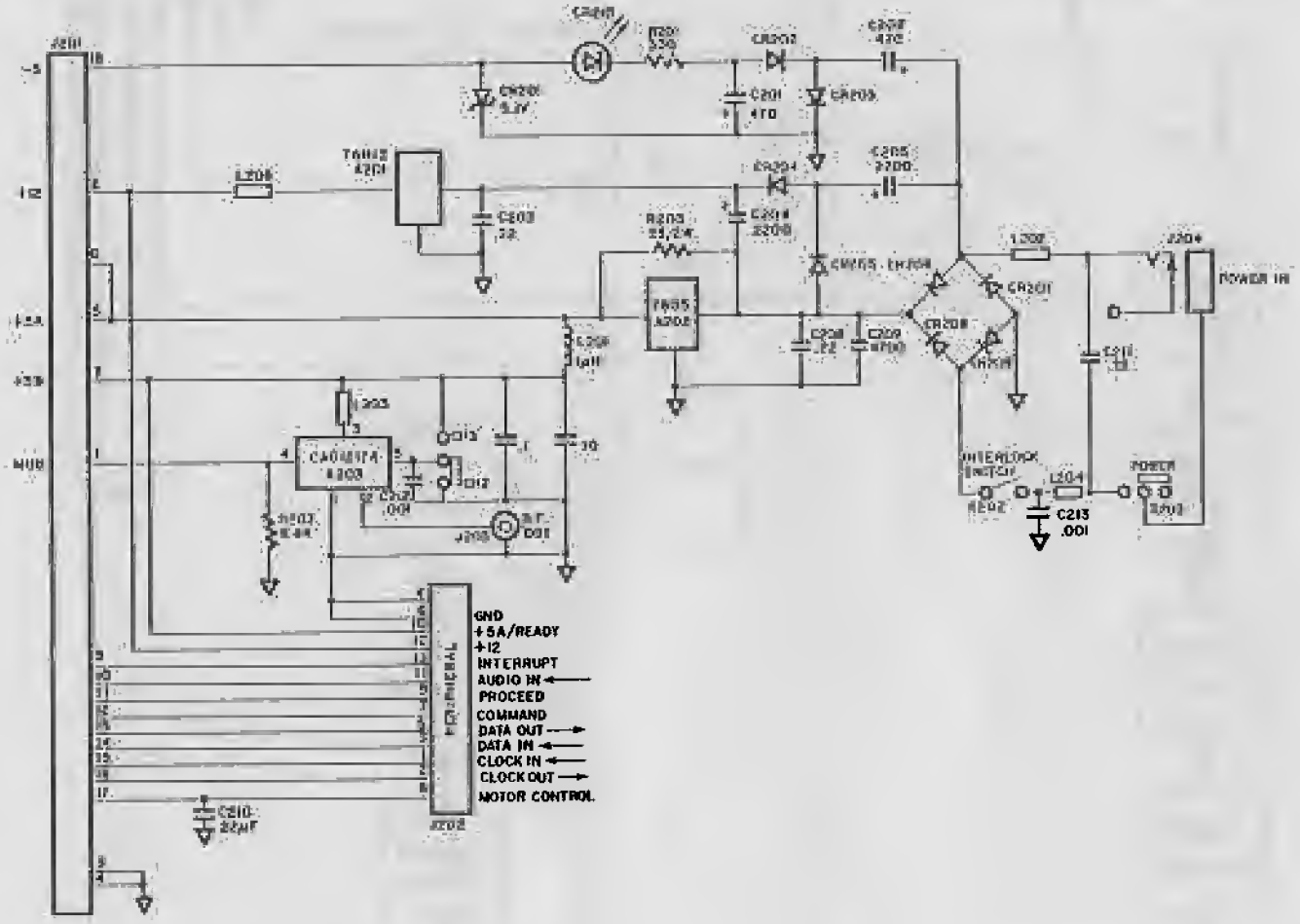
PAGE 2



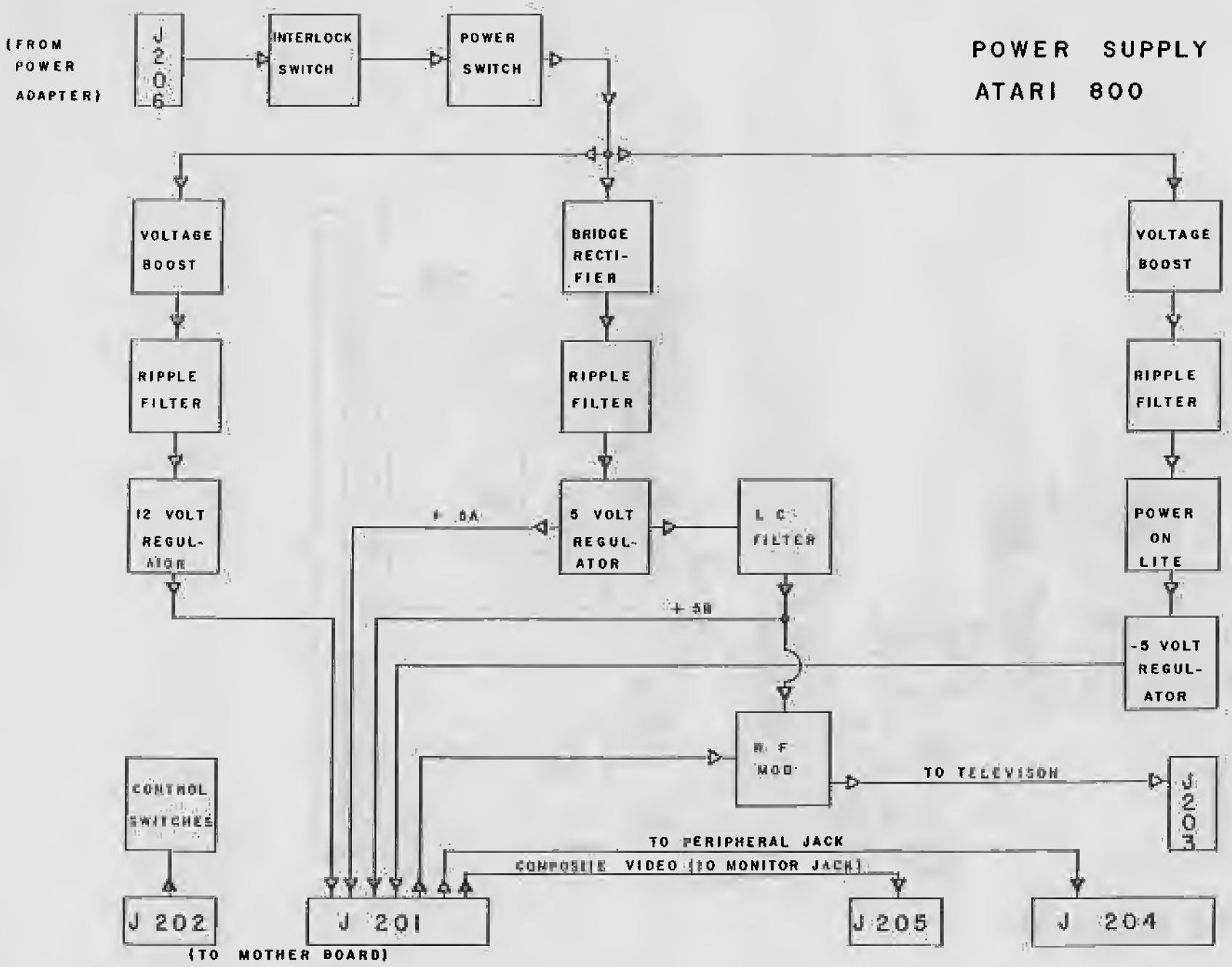
POWER SUPPLY BOARD
ATARI 400



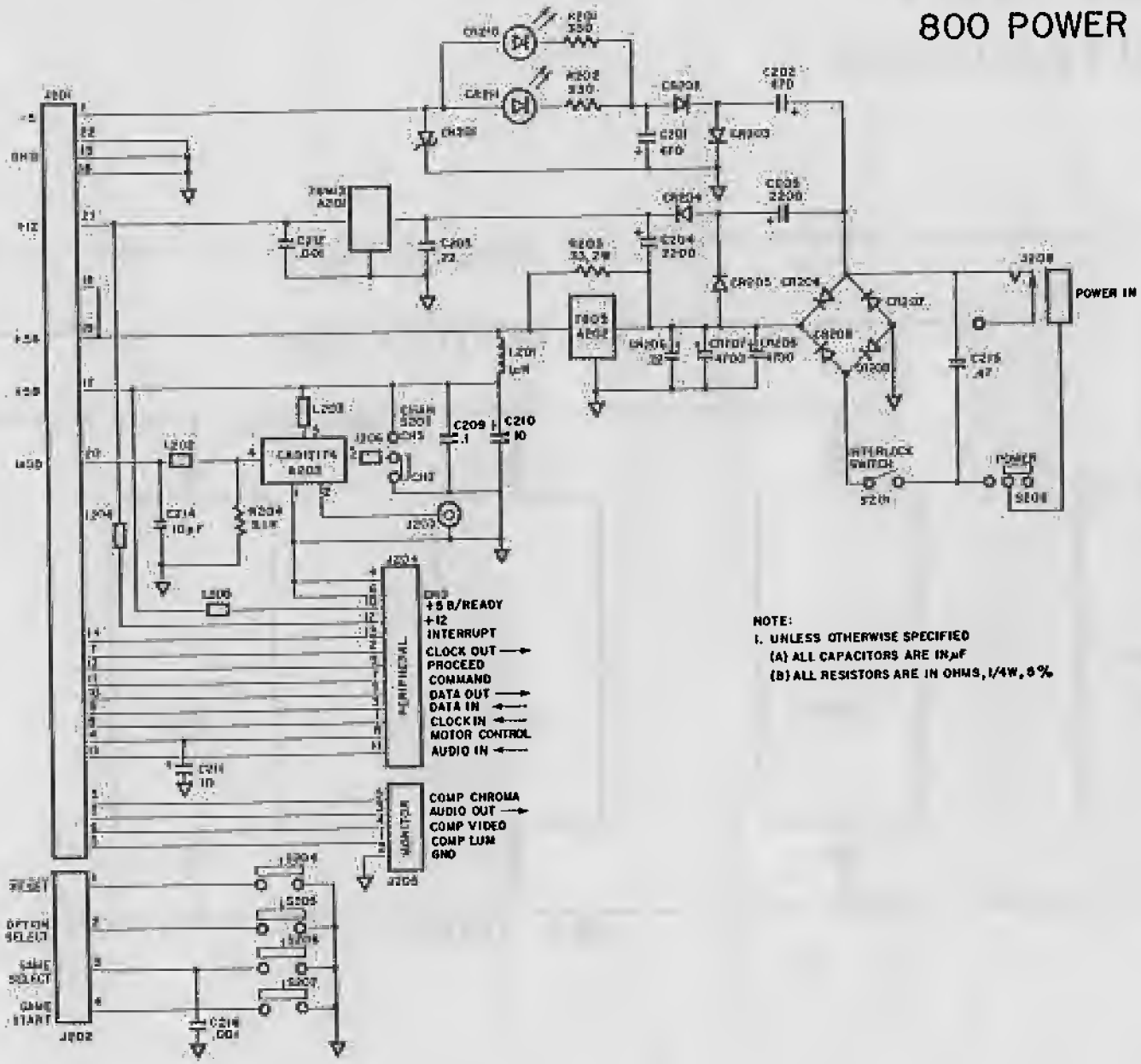
400 POWER SUPPLY



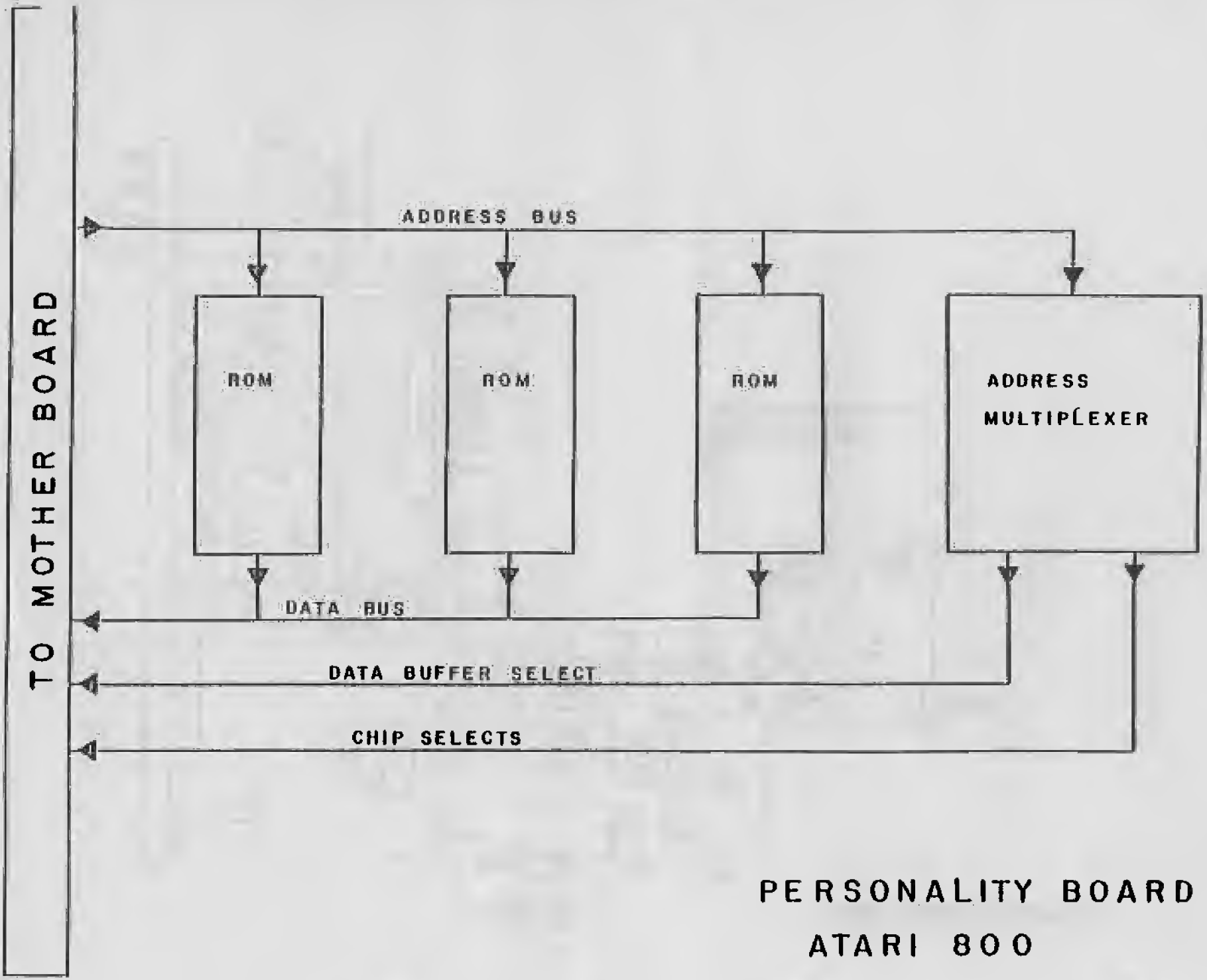
POWER SUPPLY BOARD
ATARI 800



800 POWER SUPPLY



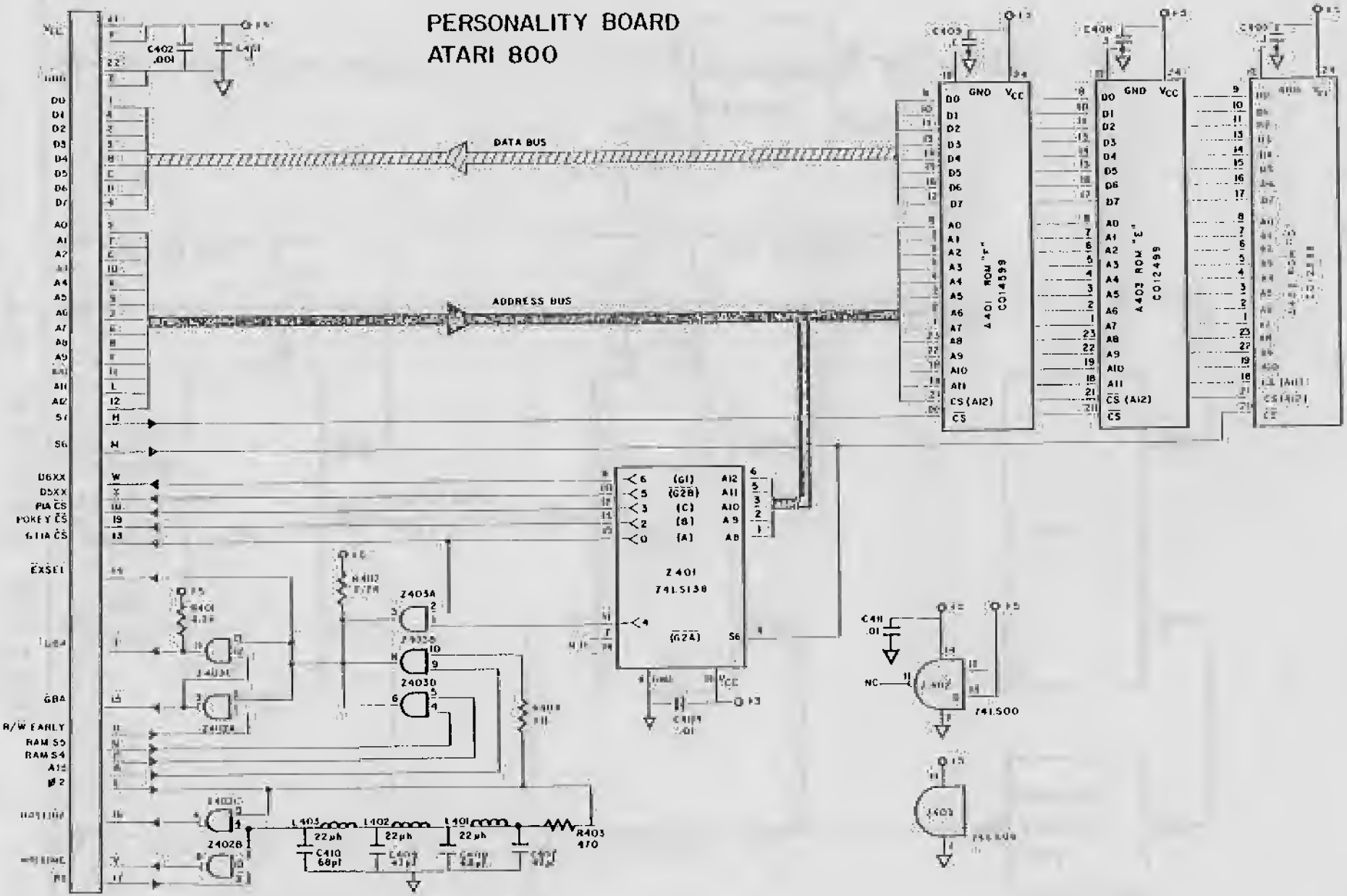
NOTE:
1. UNLESS OTHERWISE SPECIFIED
   (A) ALL CAPACITORS ARE IN μF
   (B) ALL RESISTORS ARE IN OHMS, 1/4W, 5%

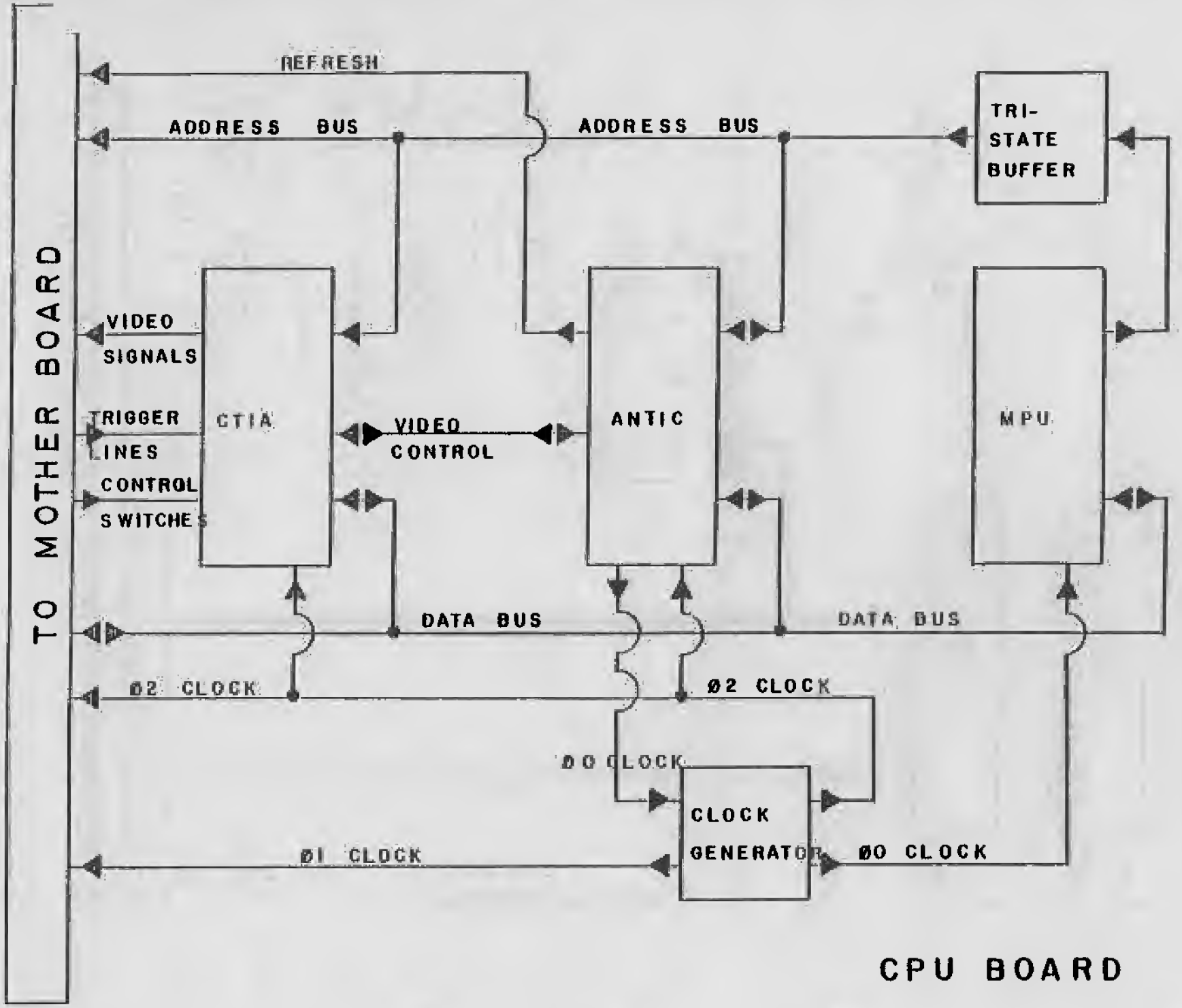


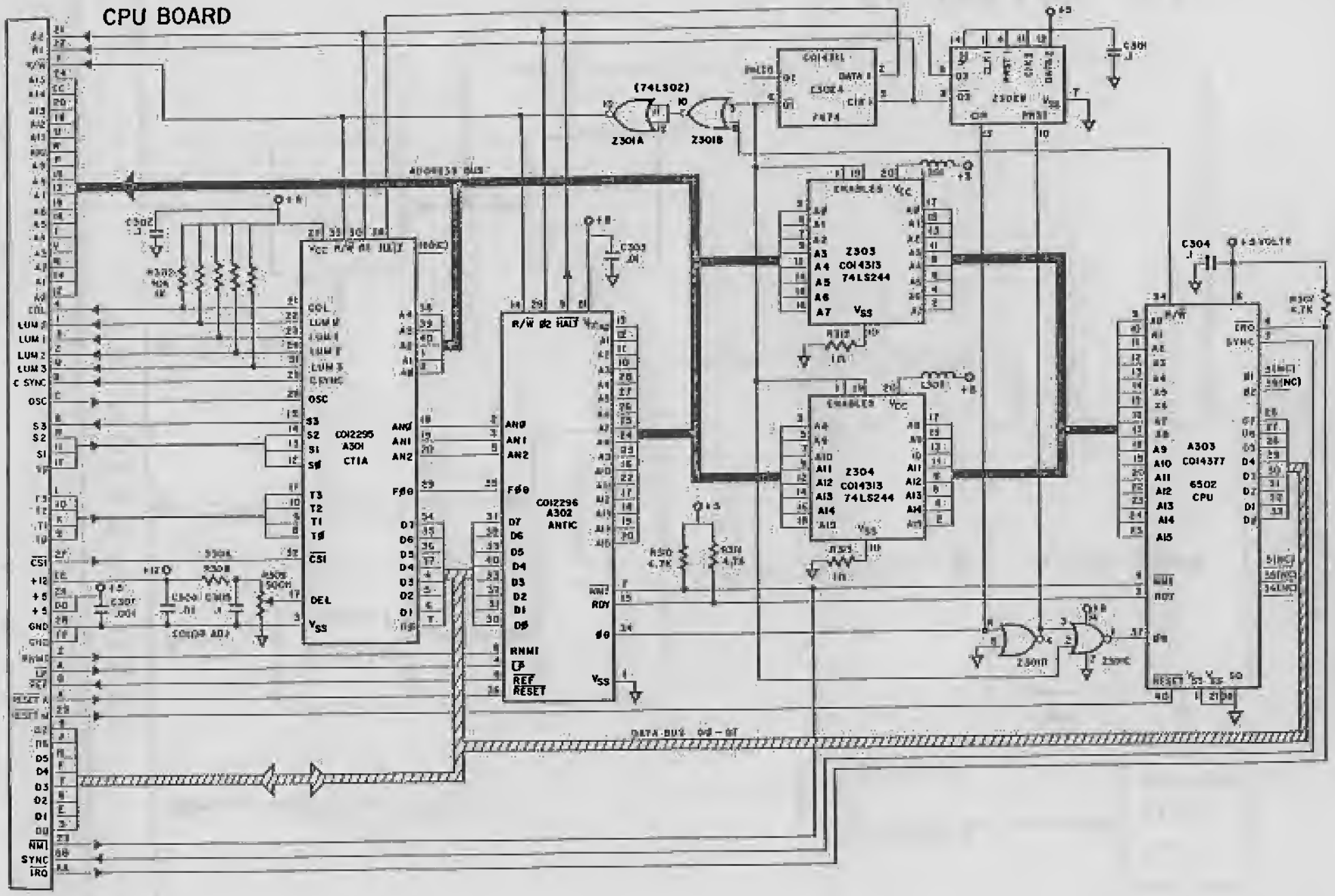
PERSONALITY BOARD

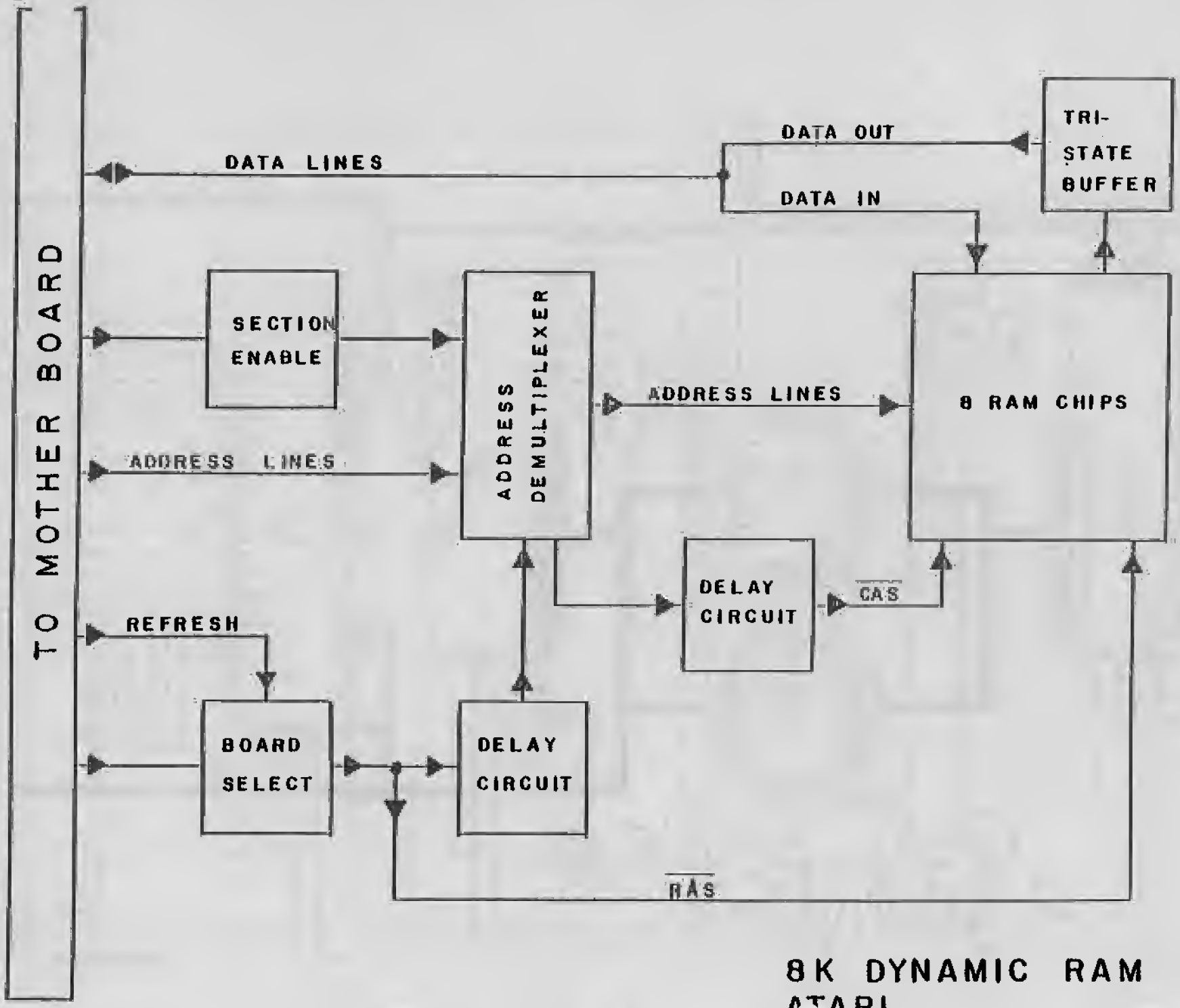
ATARI 800

PERSONALITY BOARD
ATARI 800



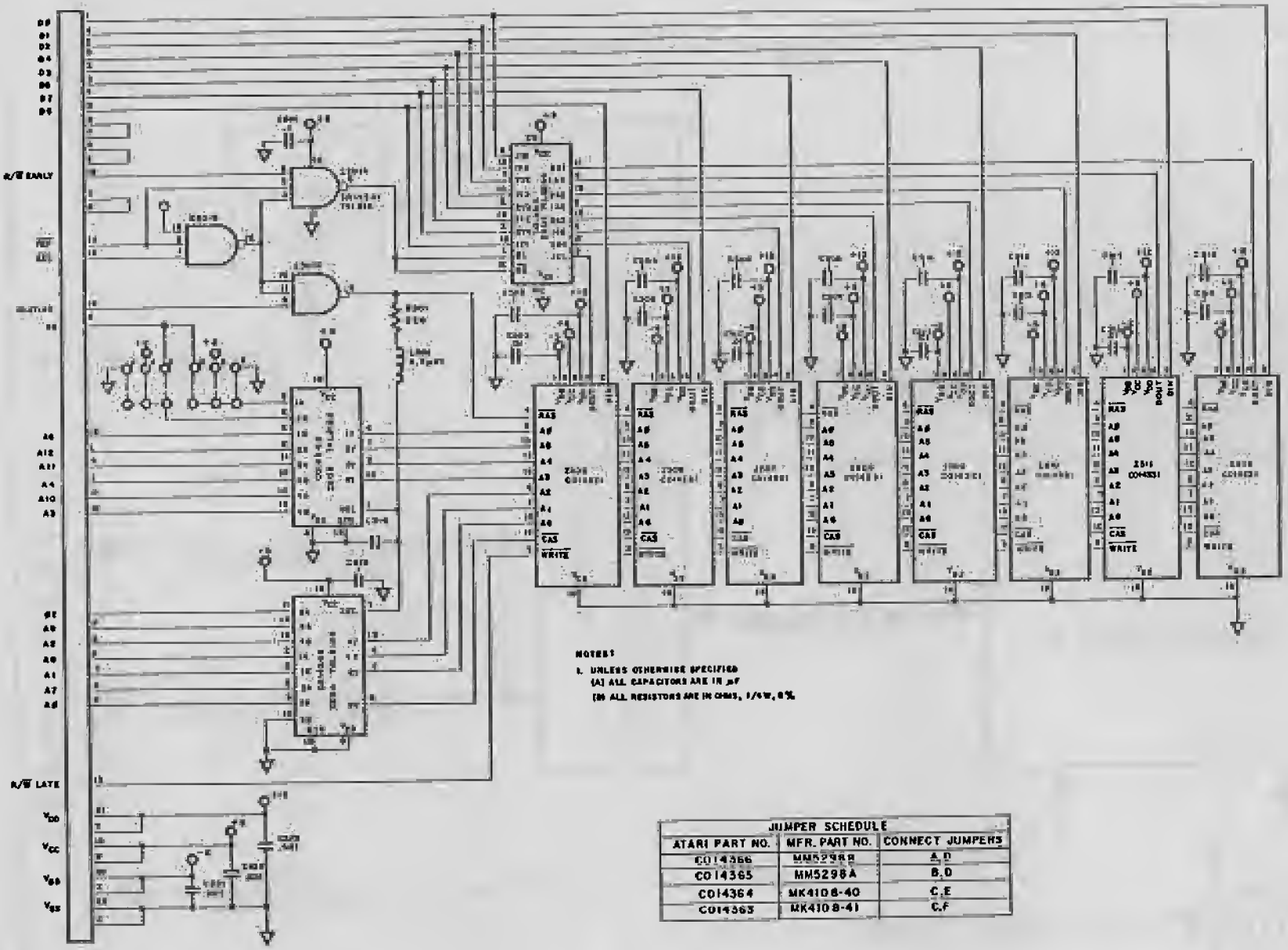

TO MOTHER BOARD
REFRESH
ADDRESS BUS
ADDRESS BUS
TRI-
STATE
BUFFER
VIDEO
SIGNALS
TRIGGER
LINES
CONTROL
SWITCHES
CTIA
VIDEO
CONTROL
ANTIC
MPU
DATA BUS
DATA BUS
Ø2 CLOCK
Ø2 CLOCK
Ø0 CLOCK
CLOCK
GENERATOR
Ø1 CLOCK
Ø0 CLOCK
CPU BOARD


CPU BOARD
(74LS302)
Z301A
Z301B
Z303
CO14313
74LS244
Z304
CO14313
74LS244
A303
CO14377
6502
CPU
CO12295
A301
CTIA
CO12296
A302
ANTIC
ADDRESS BUS
DATA BUS D0 – D7
+5 VOLTS



8K DYNAMIC RAM
ATARI

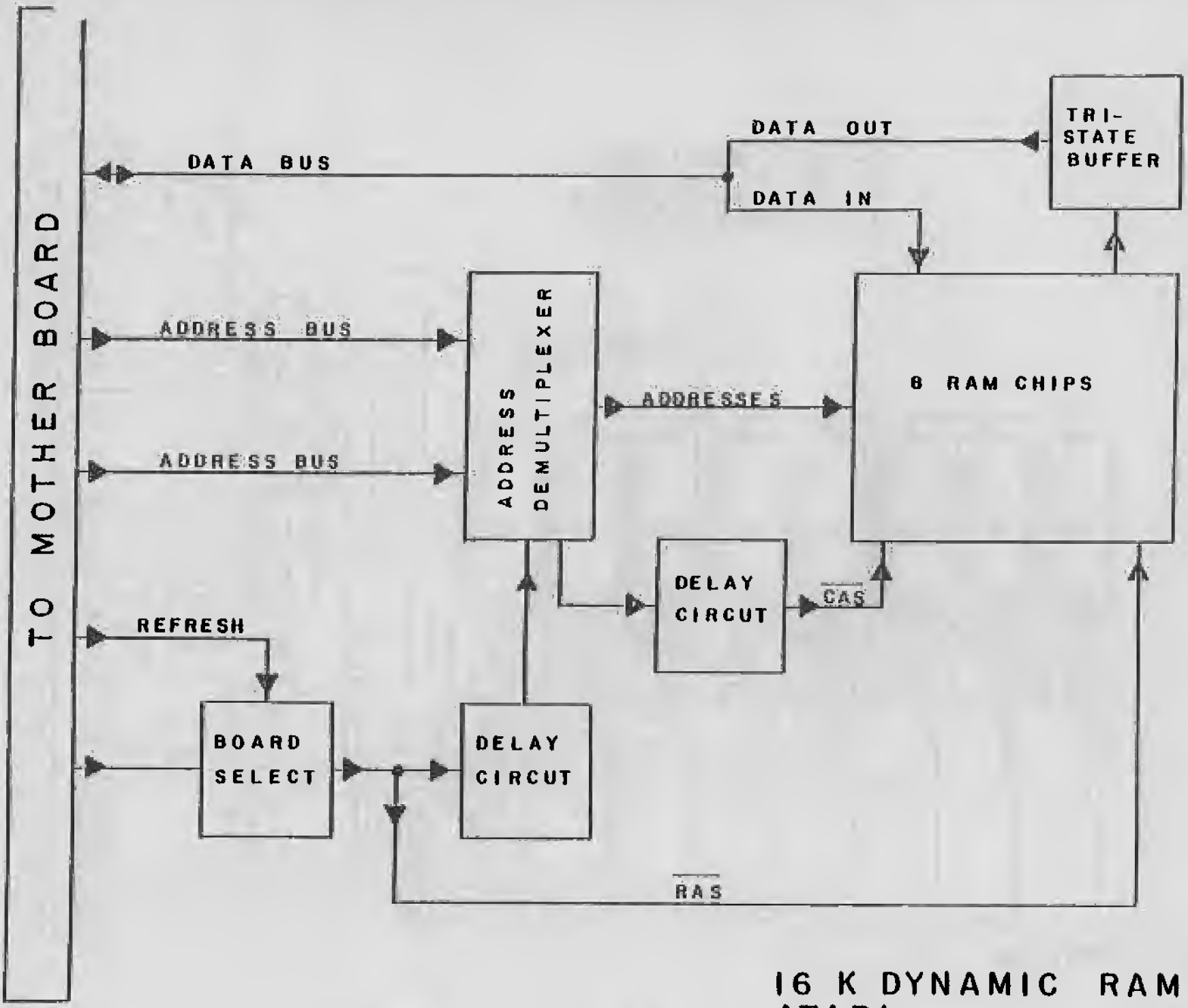
8K RAM



NOTES:

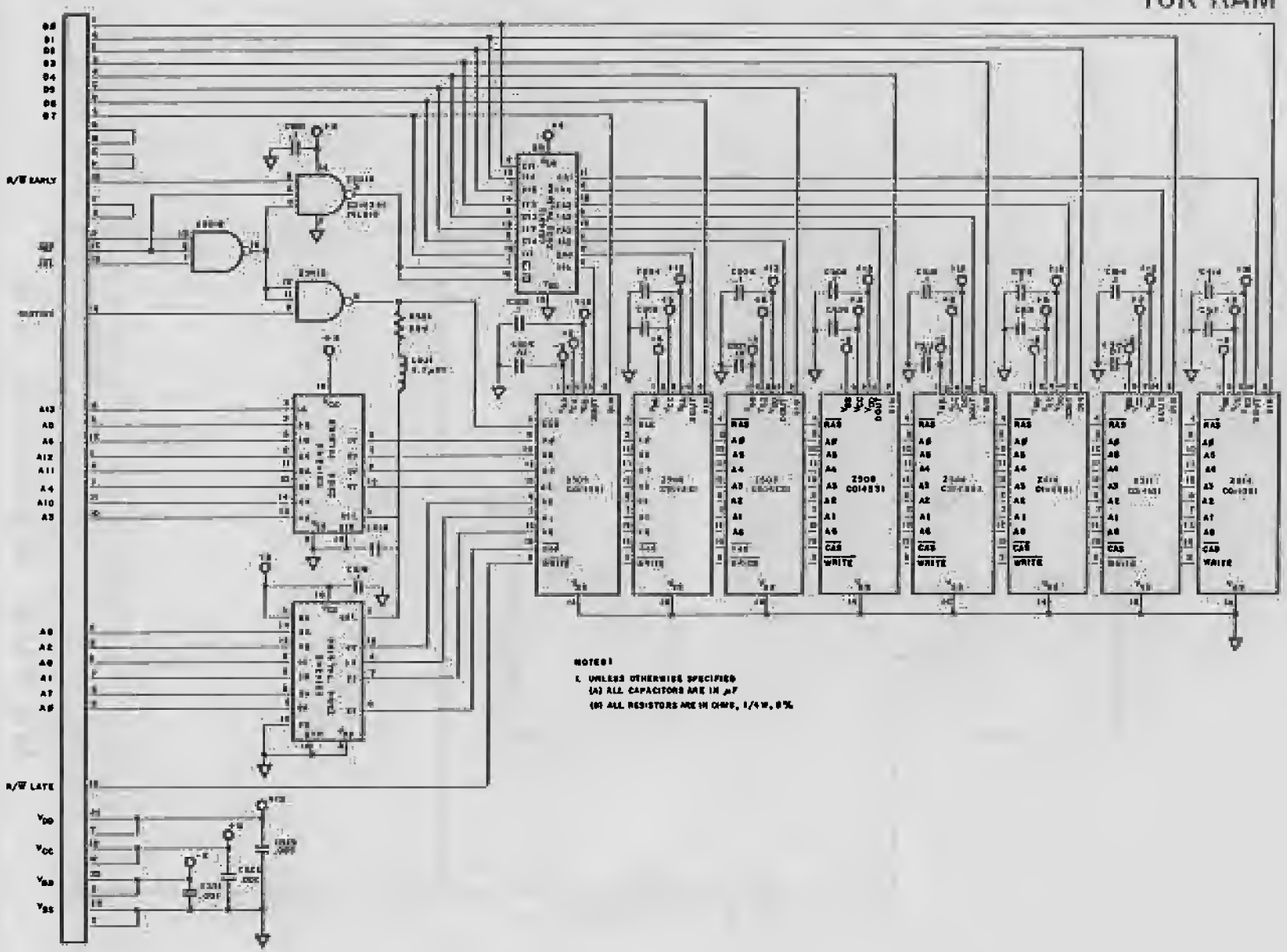
1. UNLESS OTHERWISE SPECIFIED
   (A) ALL CAPACITORS ARE IN μF
   (B) ALL RESISTORS ARE IN OHMS, 1/4W, 5%

JUMPER SCHEDULE		
ATARI PART NO.	MFR. PART NO.	CONNECT JUMPERS
C014366	MM5298B	A,D
C014365	MM5298A	B,D
C014364	MK4108-40	C,E
C014363	MK4108-41	C,F



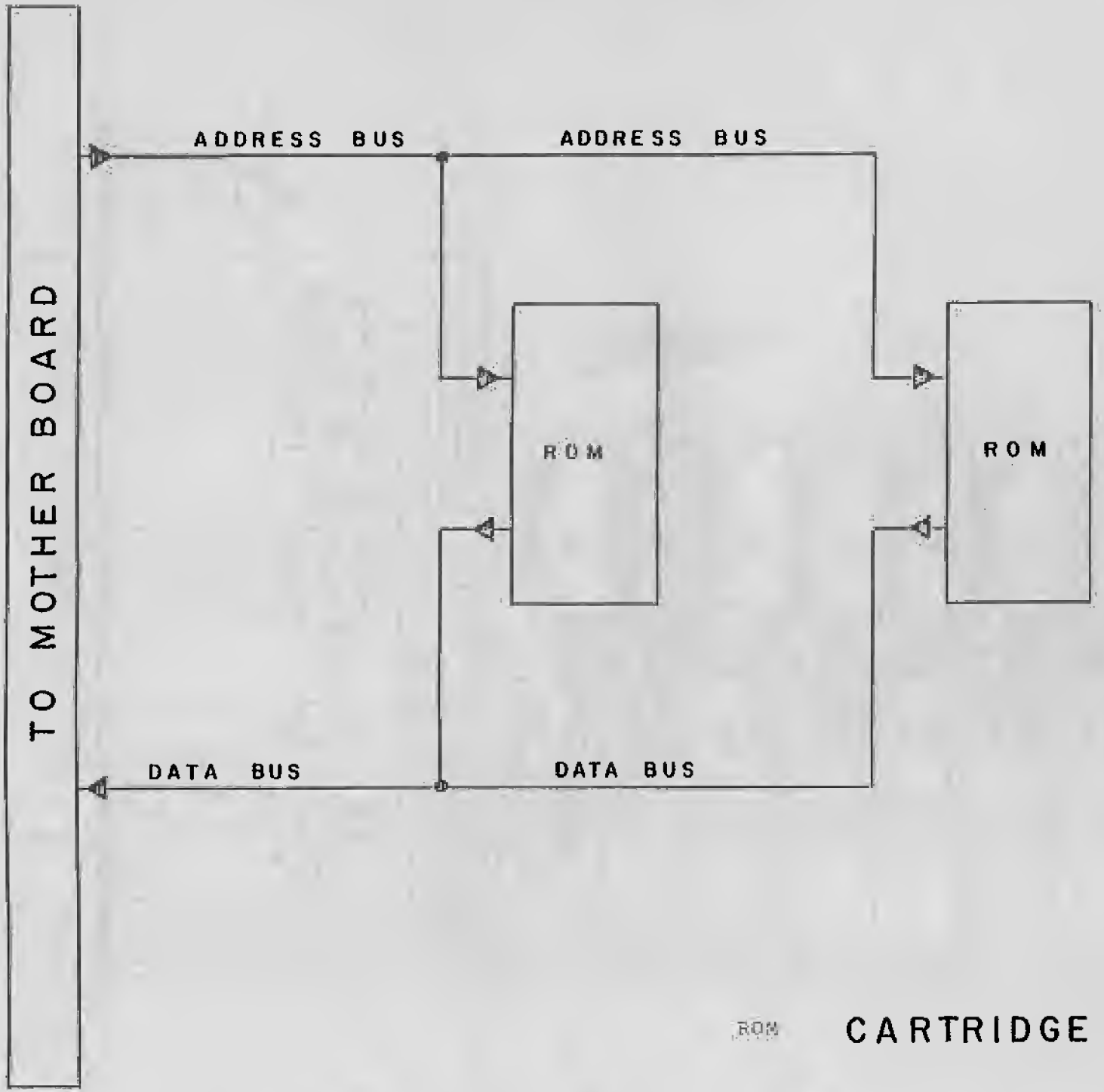
16 K DYNAMIC RAM
ATARI

16K RAM



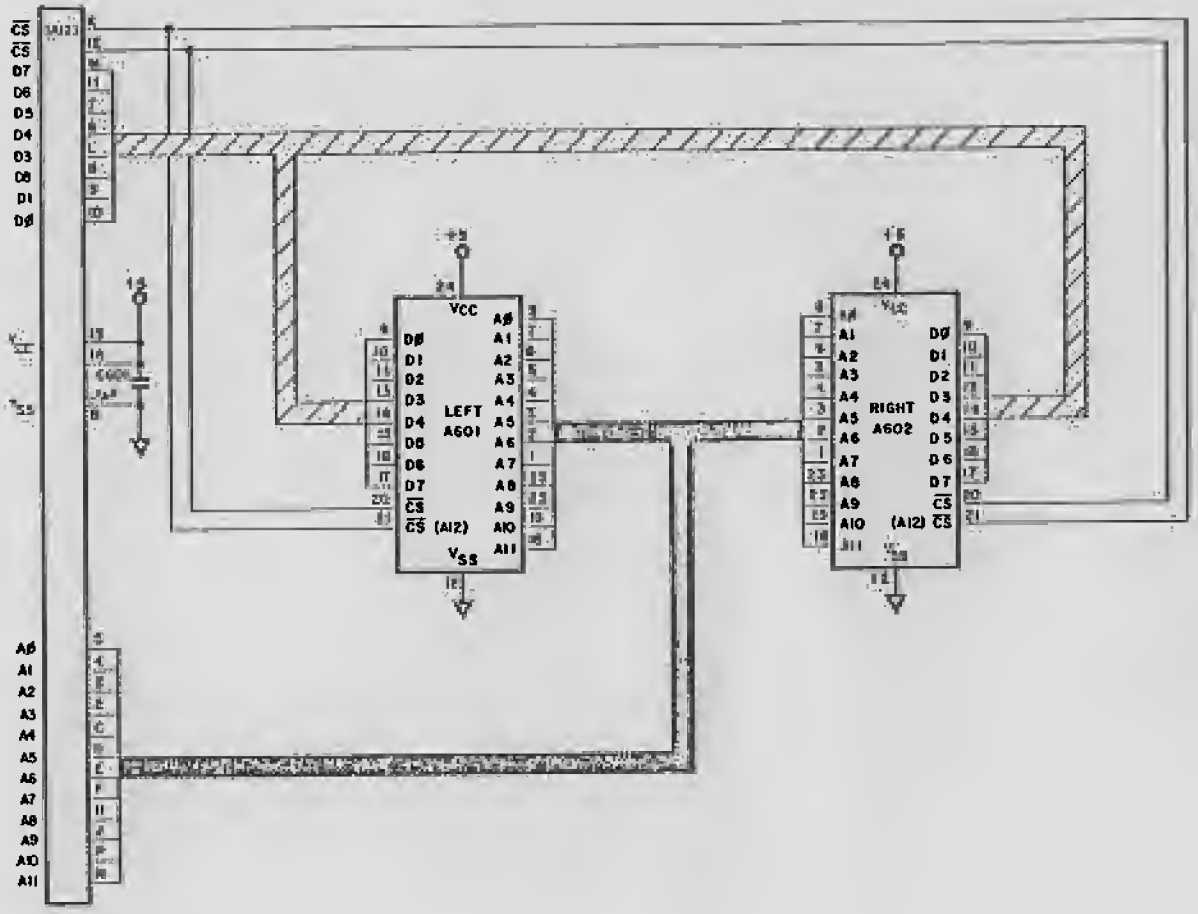
NOTES:

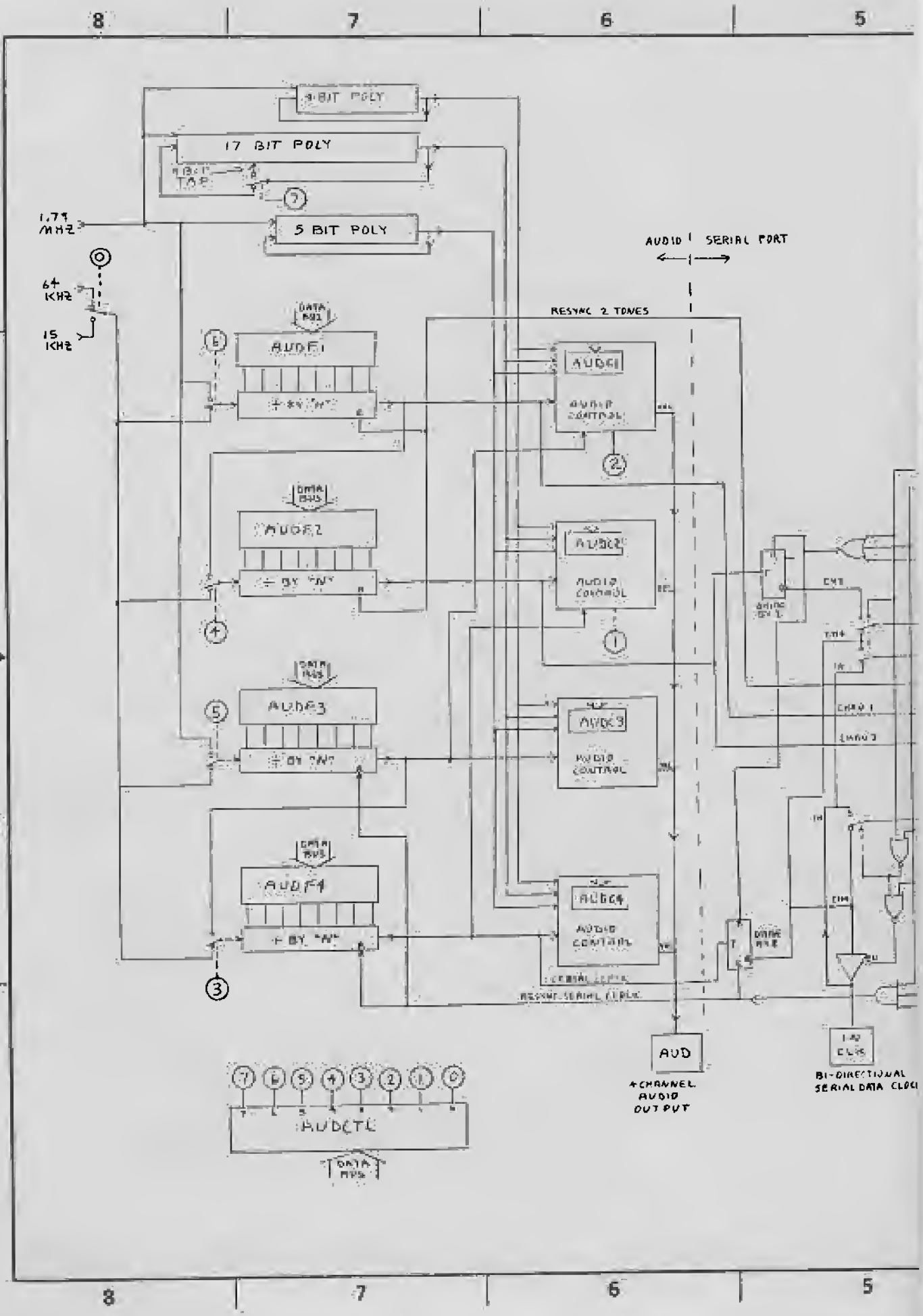
1. UNLESS OTHERWISE SPECIFIED
   (a) ALL CAPACITORS ARE IN μF
   (b) ALL RESISTORS ARE IN OHMS, 1/4 W, 5%

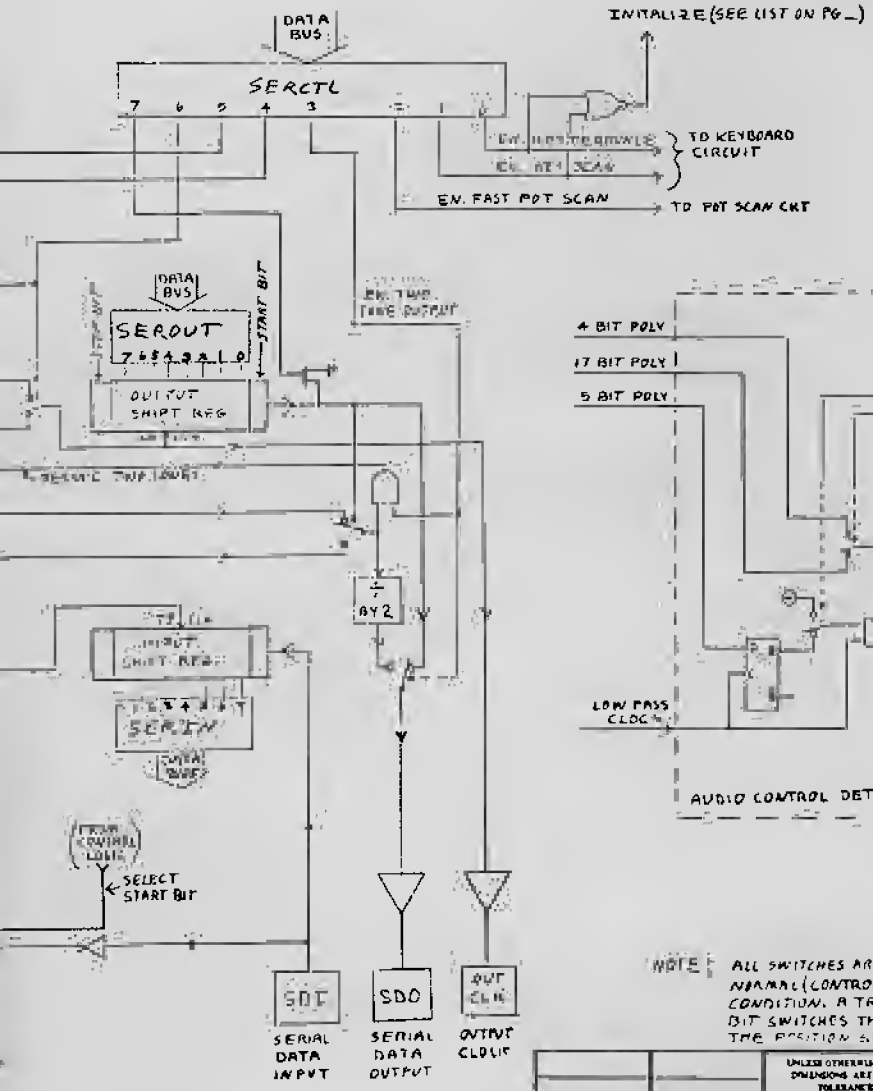


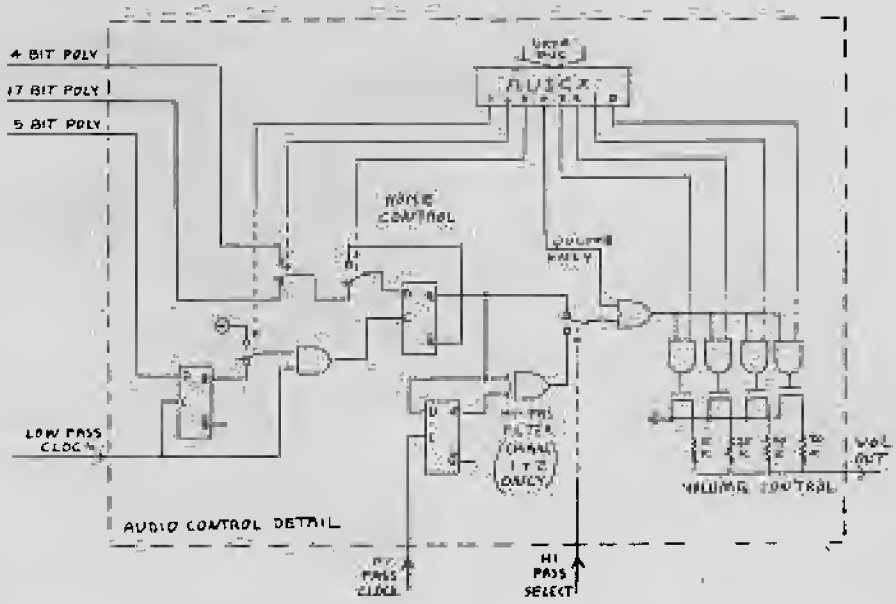
ROM CARTRIDGE BOARD

SCHEMATIC ROM CARTRIDGE




4 BIT POLY
17 BIT POLY
TAP
5 BIT POLY
1.79 MHZ
64 KHZ
15 KHZ
AUDIO SERIAL PORT
RESYNC 2 TONES
AUDF1
AUDF2
AUDF3
AUDF4
AUDCTL
AUDC1
AUDC2
AUDC3
AUDC4
AUDIO CONTROL
AUD
4 CHANNEL AUDIO OUT PUT
BI-DIRECTIONAL SERIAL DATA CLOCK





NOTE: ALL SWITCHES ARE SHOWN IN THEIR NORMAL (CONTROL REGISTER CLEARED) CONDITION. A TRUE (LOGIC 1) CONTROL BIT SWITCHES THEM AWAY FROM THE POSITION SHOWN.

OUTSTANDING ECN

1ST REV. ASSY'D	
2ND REV. ASSY'D	

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES ON
ANGLES ± 1°
.X ± .1
.XX ± .03
.XXX ± .015
SURFACE FINISH √

DO NOT SCALE DRAWING

MATERIAL

FINISH

NOTICE TO ALL PERSONS RECEIVING THIS DRAWING: CONFIDENTIAL. Reproduction or publication without the express written permission of Atari Inc., Los Gatos, California. This drawing is and remains the property of Atari Inc. and is loaned subject to its return upon demand or contract completion. Neither it nor the information it contains is to be used or disclosed except as authorized in writing by Atari Inc. The property rights to any inventions disclosed herein are retained by Atari Inc. Permission to copy this drawing does not convey a license under any patent or other property right. Copyright © 1982 Atari Inc. All rights reserved. This drawing is protected as an unpublished work under the copyright laws. In the event of publication, the following notice shall apply: Copyright © 1982 Atari Inc. All rights reserved.

ATARI

Atari, Inc.
1265 Borregas Avenue
Sunnyvale, Calif. 94086
A Warner Communications Company

TITLE: COLLEEN MANUAL
AUDIO AND SERIAL PORT
BLOCK DIAGRAM

SIZE D

SHEET _ OF _

APPENDIX A

USE OF PLAYER/MISSILE GRAPHICS

WITH BASIC

The ATARI® 400/800™ Hardware Manual should be read first to understand the details of the Player/Missile Graphics.

To enable the P/M Graphics from BASIC the following procedure can be used:*

1. Generate the playfield, either with a GRAPHICS call or build a custom display list with a series of POKE statements.

2. Enable P/M DMA control by a POKE 559 with either a 62 for single line resolution players or a 46 for double line resolution players.

3. There are four players and four missiles (or five players if the four missiles are combined into one player). Each of these has a horizontal position register that controls its horizontal position on the screen. The registers and their locations are as follows:

ADDRESS	HORIZONTAL POSITION OF
53248	Player 0
53249	Player 1
53250	Player 2
53251	Player 3
53252	Missile 1
53253	Missile 2
53254	Missile 3
53255	Missile 4

The horizontal positions can range on the playfield between 41 and 200. So POKE 53249,120 will move Player 1 to the middle of the screen.

---

*NOTE: All number references are decimal.

4. Each player (and its missile) has a color register which determines its color. These registers can be controlled by poking to the following locations:

ADDRESS	COLOR OF
704	P/M 0
705	P/M 1
706	P/M 2
707	P/M 3
711	fifth player (if enabled)

Thus a POKE 706,200 will color player 2 green.

5. The P/M bit information (those bytes which actually describe the shape of the player) must be stored in an area where it will not interfere with BASIC or the operating system. It must also start at a 2K memory boundary if single line resolution players are used, or a 1K boundary for double line resolution players.

6. The page number (i.e. number of 256 byte sections of memory) for the starting address of the P/M information obtained in step 5 is poked into location 54279.

7. Enable the P/M DMA by a POKE 53277,3.

8. The starting address of each player is obtained by multiplying the number obtained in step 6 by 256 and then adding the offset indicated in P/M memory configuration table.

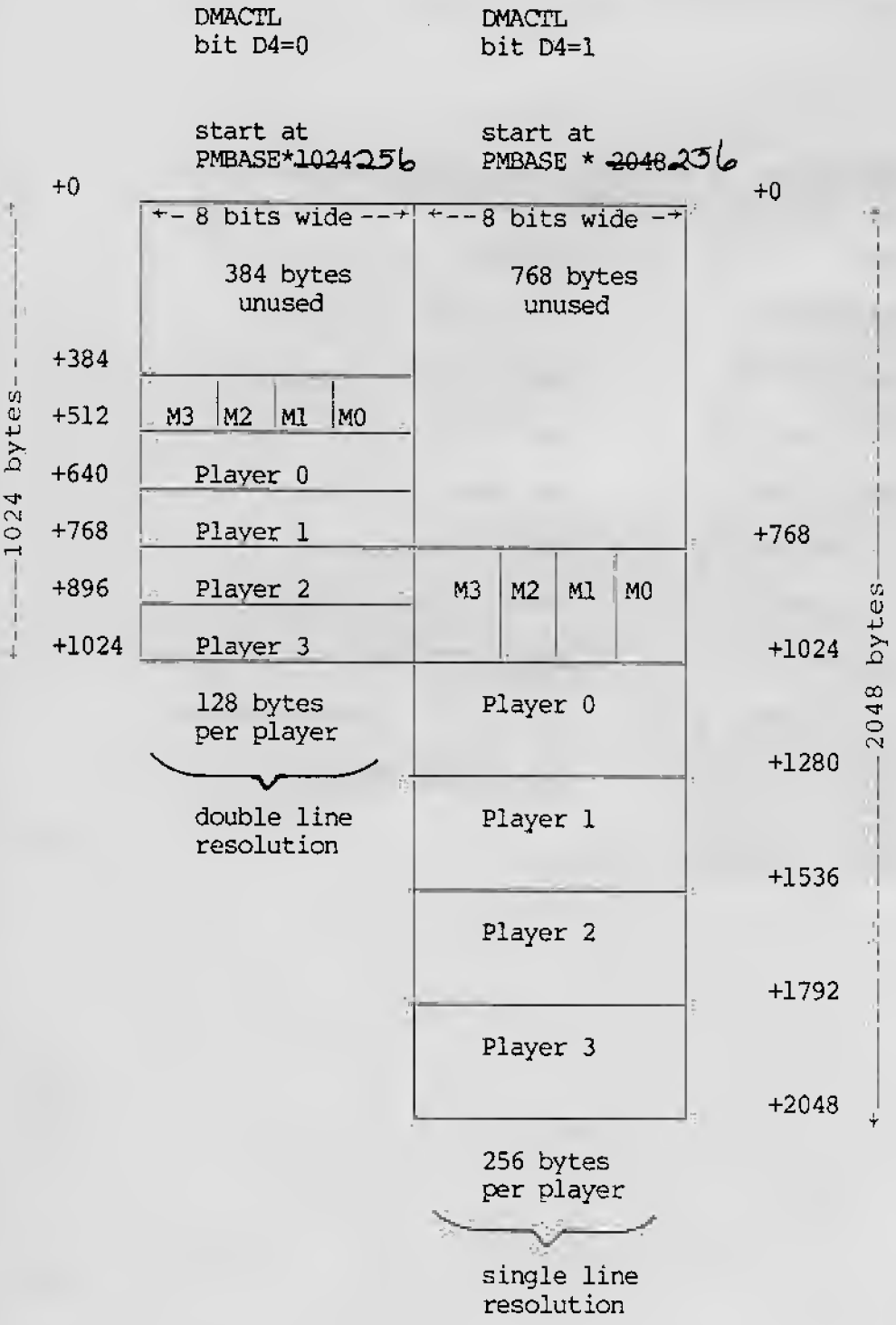
9. The vertical position of the player is determined by its location in memory. After the initial offset is obtained in step 8, its height may be defined. Its range on the playfield is from 32 to 223 in single line resolution and from 16 to 111 in double line resolution. By adding the desired height to the initial offset, the absolute address of each player is found. The appropriate bit information for the player can now be poked into this address.

(9, cont.)

Example to Generate a rectangular box player, eight color clocks wide and four lines high in immediate mode.

STEP	TYPE	RESULT
1	GRAPHICS 8	Setup Mode 8 Playfield
2	POKE 559, 62	Enable P/M DMA single line
3	POKE 53248,120	Set horizontal position
4	POKE 704,88	Set color to pink
5	I = PEEK(106)-8	Get P/M base address
6	POKE 54279,I	Store in base register
7	POKE 53277,3	Enable P/M DMA
8	J = I * 256 + 1024	Get player starting address
9	POKE J + 125,255 POKE J + 126,129 POKE J + 127,129 POKE J + 128,255	Draw player on screen

Use of Player/Missile Graphics
with BASIC, cont.



Absolute address determined by PMBASE.

Relative address shown along sides of maps.

Each Player-Missile section (128 bytes in single line, 256 bytes in double line) maps directly onto the total height of TV screen.

APPENDIX B

MIXING GRAPHICS MODES

I. GENERAL

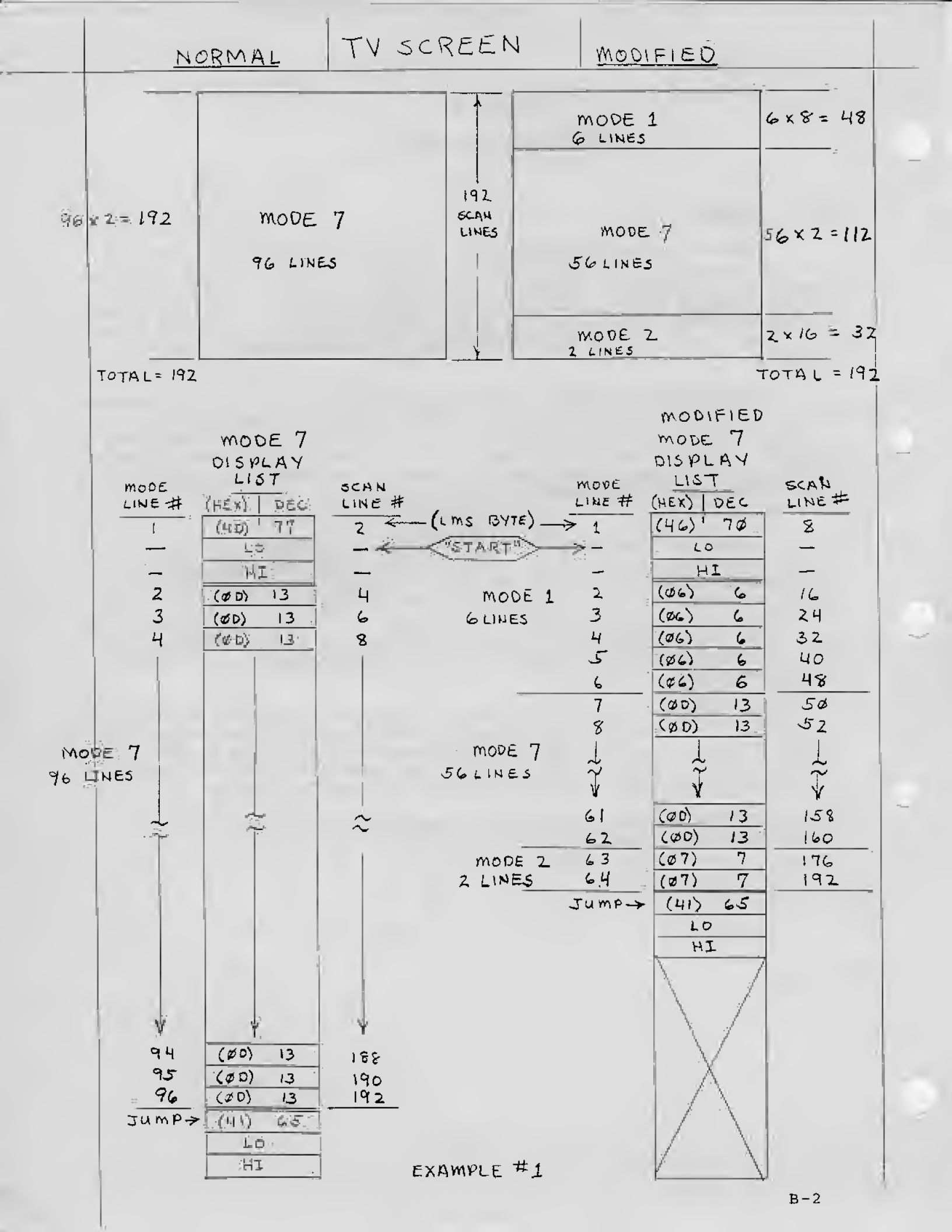
This procedure describes how to mix several graphic modes on the TV screen at the same time using BASIC commands. Each graphics mode has a different number of scan lines per "Mode Line" (one line of a graphics mode). The TV screen must consist of 192 scan lines, so when mixing modes, they must be combined in such a way as to get 192 scan lines. This is accomplished by modifying the Display List.

When a graphics mode is set on the computer, the O/S allocates RAM space for the graphics mode, then builds the display list adjacent to the graphics RAM, and sets a pointer to the beginning of the display list. Each "mode line" is constructed from a "mode byte" in the display list that determines how many scan lines in each mode line. The display list describes the screen display from top to bottom.

A Display List must be built for the "max RAM mode" (the graphics mode that requires the most RAM) then modified with POKES to mix the other modes with it. This "max RAM mode" cannot be a split screen mode (text window), therefore "max RAM mode" +16 must be used. If the max RAM mode will be at the top of the screen, then the "LMS byte" (load memory scan byte) at the top of the Display List will already be correct. If not, the "LMS byte" will have to be modified.

The Display List is modified by POKING a new mode byte for each mode line that is <u>not</u> a max RAM mode line. At the end of the display list is a JUMP instruction pointing to the top of the Display list. When the Display List is modified, the JUMP instruction must be placed immediately after the last mode byte.

Example #1 will be used throughout this procedure to illustrate each step.


NORMAL
TV SCREEN
MODIFIED
96 x 2 = 192
MODE 7
96 LINES
192 SCAN LINES
MODE 1
6 LINES
6 x 8 = 48
MODE 7
56 LINES
56 x 2 = 112
MODE 2
2 LINES
2 x 16 = 32
TOTAL = 192
TOTAL = 192
MODE 7 DISPLAY LIST
MODE LINE #
(HEX) | DEC
SCAN LINE #
(LMS BYTE)
"START"
MODIFIED MODE 7 DISPLAY LIST
MODE LINE #
(HEX) | DEC
SCAN LINE #
MODE 7
96 LINES
MODE 1
6 LINES
MODE 7
56 LINES
MODE 2
2 LINES
Jump
Jump
EXAMPLE #1

II. PROCEDURE TO SET UP SCREEN IN MIXED MODES:

1. Select modes desired, then look up which mode is the max RAM mode from table #2.

   example: modes selected - mode 1, mode 7, mode 2

   mode 7 = max RAM mode

2. Use table #1 to calculate the number of mode lines such that the total number of scan lines = 192.

   example:

mode	# mode line	scan lines per mode line	scan lines
1	6	8	48
7	56	2	112
2	2	16	32
			192 TOTAL

3. If the max RAM mode is at the top of the screen, then skip this step: Calculate the LMS byte by setting the left nibble to 4, then use table #1 to find the right nibble for the graphics mode at the top of the screen.

   example: 1. left nibble = 4
   2. right nibble for mode 1 = 6
   3. LMS byte = 46 (HEX)

4. Calculate the mode byte for each mode. Set the left nibble to 0, use table #1 to find the right nibble for each mode.

   example:

Mode	Left Nibble	Right Nibble	Mode Byte (HEX)
1	0	6	06
7	0	D	0D
2	0	7	07

II. PROCEDURE TO SET UP SCREEN IN MIXED MODES, cont.:

5. Convert all bytes to decimal.

example:

Byte	(HEX)	DEC
LMS	46	70
Mode 1	06	6
Mode 7	0D	13
Mode 2	07	7

6. Execute a graphics call on the computer using the max RAM mode (+16).

   example: GRAPHICS 7 + 16

7. PEEK the Display List pointer and use it to calculate a variable labelled "START".

   example: START = PEEK(560) + PEEK(561) * 256 + 4

8. If the max RAM mode is at the top of the screen, then skip this step: Poke the LMS byte to location START-1.

   example: POKE START-1,70

9. Every mode line requires a mode byte in the Display List in the same order as the mode lines appear on the screen. The mode bytes must be POKED into the Display List at location START + offset, where offset = mode line #.

Example:	MODE LINE #	POKE INSTRUCTION
	2	POKE START + 2,6
	3	POKE START + 3,6
MODE 1	4	POKE START + 4,6
	5	POKE START + 5,6
	6	POKE START + 6,6
MODE 7	see note for mode 7 (max RAM mode)	
MODE 2	63	POKE START + 63,7
	64	POKE START + 64,7

NOTE: The Display List will already be correct for the max RAM mode, therefore its mode bytes do not need to be POKEd.

II. PROCEDURE TO SET UP SCREEN IN MIXED MODES, cont.:

10. POKE the JUMP instruction followed by the LO byte, then the HI byte into the Display List. The offset for the JUMP POKE is the last mode line # + 1, for LO byte it is + 2, for HI byte it is + 3.

example: (last mode line # was 64)

REMARK	POKE INSTRUCTION
JUMP	POKE START + 65,65
LO BYTE	POKE START + 66, PEEK(560)
HI BYTE	POKE START + 67, PEEK(561)

III. PROCEDURE TO PRINT AND PLOT IN MIXED MODES

1. If the mode line #'s of a mode on the screen fall within the range of that mode's normal mode line #'s then use the following procedure:
   a. POKE 87 with the mode #
   b. Determine the Y coordinate by counting the # of mode lines from the top of the screen to the current position.
   c. Determine the X position in the normal manner for that mode.
   d. Depending on the mode, either PLOT and DRAWTO, or POSITION and PRINT.
   e. These steps must be done for each mode on the screen that meets the condition in step 1.

```
example: MODE 1   POKE 87,1
                  POSITION 2,1:PRINT #6;"TEXT"

         MODE 7   POKE 87,7
                  COLOR 1:PLOT 20,20:DRAWTO 30,30

         MODE 2   See step 2
```

III. PROCEDURE TO PRINT AND PLOT IN MIXED MODES, cont.

2. Some modes may have mode line #'s outside of their normal range.

example: Mode 2 normally has mode line #'s 1 through 12 (full screen). These are modified to #63 and #64 in example #1.

To prevent the computer from giving a "cursor out of range" error message the following procedure can be used:

a. Set a variable labelled "MEMST" to be the display memory start pointer.

```
MEMST = PEEK(START) + PEEK(START + 1) * 256
```

b. Set a variable labelled CHRPOS to position characters to be printed on the target line.

$$CHRPOS = MEMST + [(M_1 - 1)*R - M_2*(R-20) - M_3*(R-10)] + X$$

Where:

X = horizontal position of character on the target line.
R = the RAM per line of the Max RAM Mode (table #1).
M_1 = the Mode Line # of the target line.
M_2 = the number of mode lines of 20 bytes of RAM per line <u>above</u> the target line.
M_3 = the number of mode lines of 10 bytes of RAM per line <u>above</u> the target line.

Example: calculate CHRPOS for Mode Line #64 (the last line of the Mode 2 area) at horizontal position 5.

X = 5
R = 40
M_1 = 64
M_2 = 7 (6 from Mode 1 area, 1 from Mode 2 area).
M_3 = 0

```
CHRPOS = MEMST + [(64-1)*40-7*(40-20)-0*(40-10)]+5
CHRPOS = MEMST + [(63)*40-7*(20)-0*(30)]+5
CHRPOS = MEMST + [2520 - 140] + 5
CHRPOS = MEMST + [2380] + 5
CHRPOS = MEMST + 2385
```

III. PROCEDURE TO PRINT AND PLOT IN MIXED MODES, cont.

2. cont.

c. If few characters will be printed, then each character's internal value may be looked up in the Internal Character Set Table (Table 9.6), in the new BASIC Reference Manual. This value is then POKED into CHRPOS.

d. If strings are to be output, and if the ATASCII values of all the characters lie within one of the ranges shown in the table below, then do the following:

1) Obtain the appropriate ATASCII value range for the characters
2) Do the OPERATION the table indicates on the ATASCII value of each character.
3) POKE this value into CHRPOS.

ATASCII VALUE RANGE	OPERATION
0-31	Value + 64
32-95	Value - 32
96-127	NONE
128-159	Value + 64
160-223	Value - 32
224-255	NONE

Example: 1) assume we want to print the word "TEXT" in the mode 2 area of example #1 using the CHRPOS calculated previously.

2) these characters are in the ATASCII VALUE RANGE of "32 - 95".

3) the OPERATION for this range is "Value-32", so 32 must be subtracted from each ATASCII value.

4) the program statements would now look like this:

```
T$(1,4) = "TEXT"
CHRPOS  = MEMST + 2385

FOR X = 1 TO LEN(T$)
POKE CHRPOS + X - 1, ASC[T$(X,X)] - 32

NEXT X
```

(OPERATION: value - 32)

5) the FOR/NEXT loop POKEs the first character of T$, ASC[T$(X,X)]-32, into CHRPOS + 0.

6) the next iteration POKEs the next character of T$ into the next CHRPOS, and so on.

TABLE #1

MODE BYTE

REMARK	LEFT NIBBLE (HEX)		RIGHT NIBBLE (HEX)	C.C. PER PIXEL	SCAN LINES PER MODE LINE	# COLORS	MODE	RAM PER LINE
			2	½	8	1½ ③	0	40
①	4	CHAR	3	½	10	1½	-	40
			4	1	8	4	-	40
		MODES	5	1	16	4	-	40
②	0		6	1	8	5	1	20
			7	1	16	5	2	20
			8	½	8	4	3	10
			9	½	4	2	4	10
①	4	GRAPHIC	A	½	4	4	5	20
			B	1	2	2	6	20
		MODES	C	1	1	2	-	20
			D	1	2	4	7	40
②	0		E	1	1	4	-	40
			F	1	1	1½	8	40
BLANK	0-7	④	0	BLANK	-	-	-	-
JUMP	4	SPECIAL	1	JUMP	-	-	-	-

① When the max RAM mode is not at the top of the screen, the left nibble of the LMS byte must be changed to a 4.

② Left nibble for all mode bytes after the LMS byte.

③ Color & Lum for the field is controlled by Setcolor 2, and Lum for charactors or graphics from Setcolor 1.

④ JUMP - used to end the display list and return to the beginning.

BLANK - to output selected number of background lines.

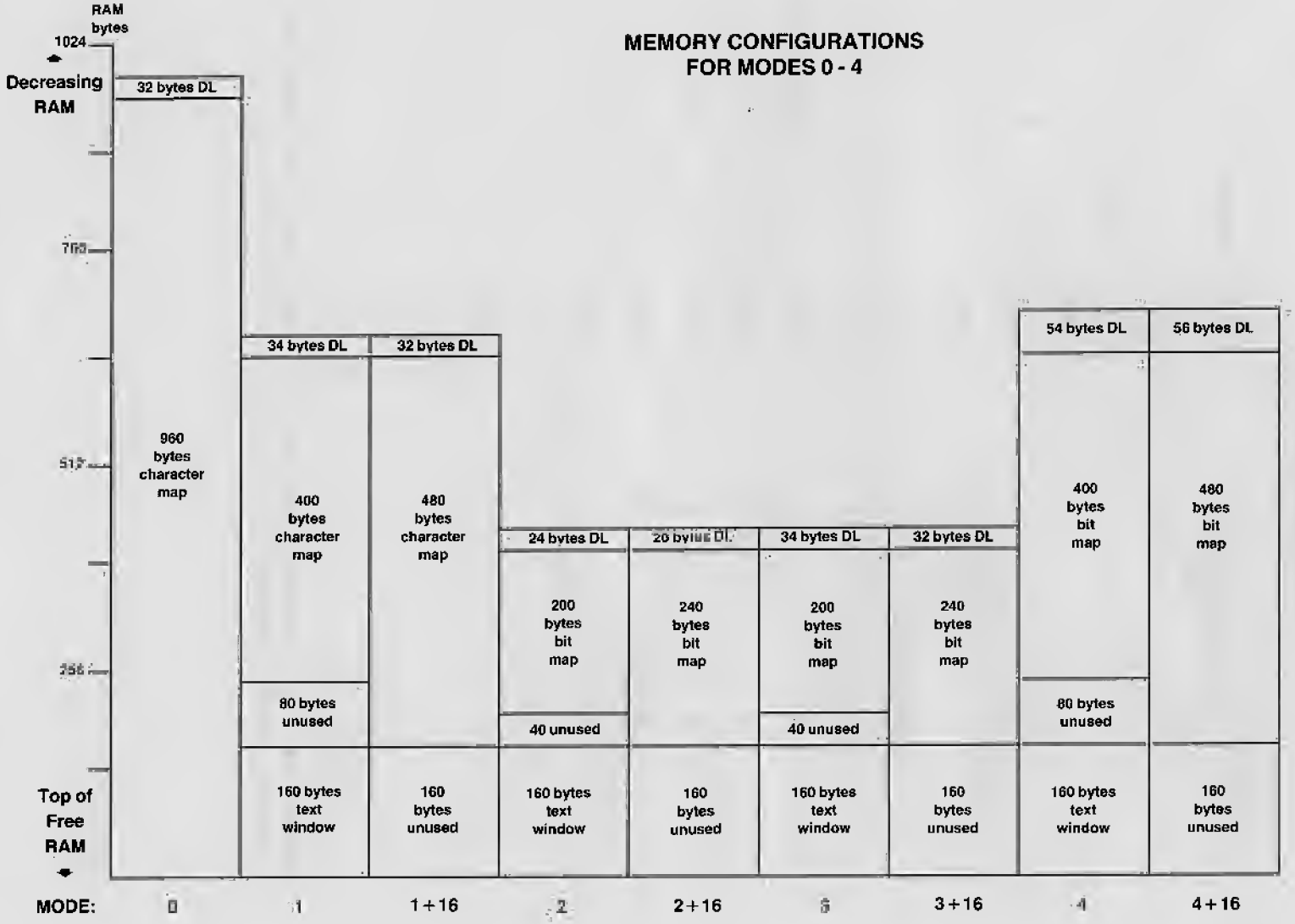
TABLE #2

GRAPHICS MODES
RAM REQUIREMENTS

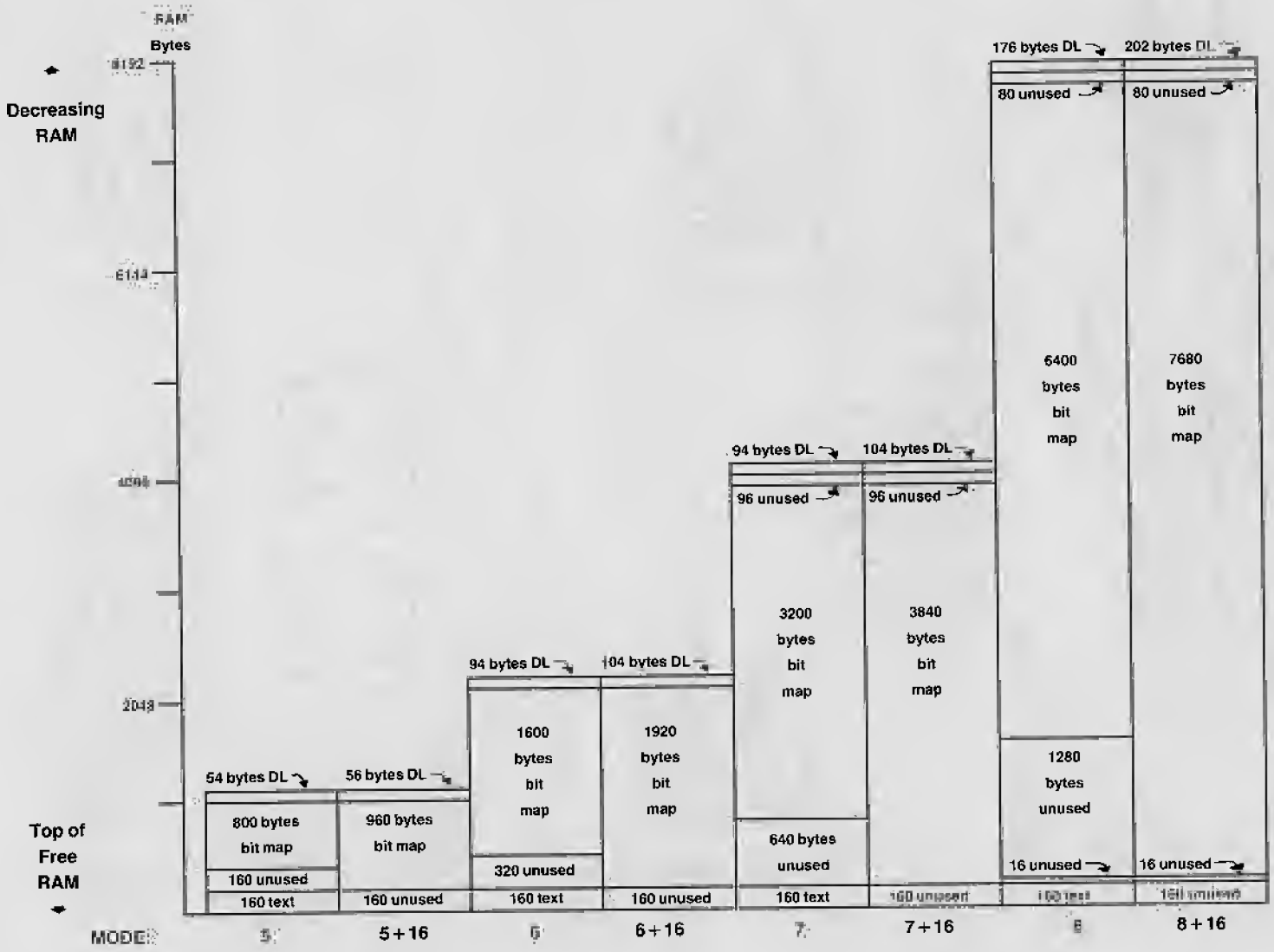
Mode		
	8 + 16	8138 Bytes
	8	8112
	7 + 16	4200
	7	4190
	6 + 16	2184
	6	2174
	5 + 16	1176
	5	1174
	4 + 16	696
	4	694
	3 + 16	432
	3	434
	2 + 16	420
	2	424
	1 + 16	672
	1	674
	0	992

These values include the display list and any imbedded unused memory blocks.

MEMORY CONFIGURATIONS FOR MODES 0 - 4



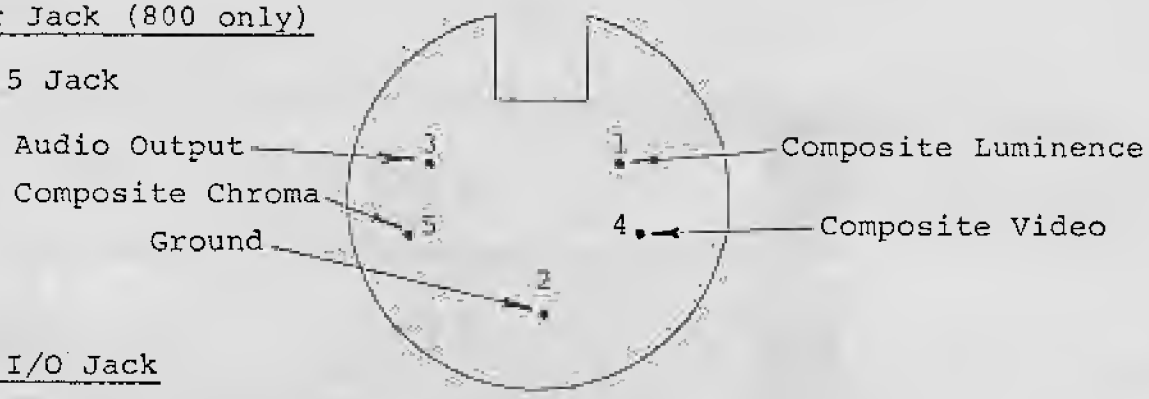
MEMORY CONFIGURATIONS FOR MODES 5 - 8



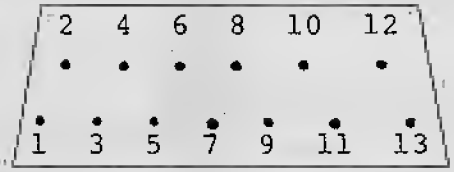
APPENDIX C: PINOUTS

Monitor Jack (800 only)

D.I.N. 5 Jack

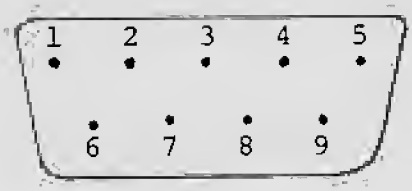


Serial I/O Jack



1. Clock Input
2. Clock Output
3. Data Input
4. Ground
5. Data Output
6. Ground
7. Command
8. Motor Control
9. Proceed
10. +5/Ready
11. Audio Input
12. +12 volts
13. Interrupt

Controller Jack



1. (Joystick) Forward Input
2. (Joystick) Back Input
3. (Joystick) Left Input
4. (Joystick) Right Input
5. B Potentiometer Input
6. Trigger Input
7. +5 volts
8. Ground
9. A Potentiometer Input